/N-02
VOL.2
19377
P.71

NASA Contractor Report 187484, Volume II

A General Multiblock Euler Code for Propulsion Integration,

Volume II: User Guide for BCON, Pre-Processor for Grid Generation and GMBE

T. Y. Su, R. A. Appleby, and H. C. Chen,

Boeing Commercial Airplane Group
Seattle, Washington

Contract NAS1-18703
May 1991

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

CONTENTS

1. SUMMARY

BCON is a menu-driven graphics interface program. The BCON input consists of strings or arrays of points generated from a CAD tool or any other surface geometry source. The user needs to design the block topology, prepare the surface geometry definition and surface grids separately. BCON generates input files, that contain the block definitions and the block relationships required for generating a multi-block volume grid with the EAGLE grid generation package. BCON also generates the block boundary conditions file which is used along with the block relationship file as input for the General MultiBlock Euler code (GMBE, Volumes I and III).

2. INTRODUCTION

The interactive graphics interface program BCON is a menu-driven program for preparing block definition, block-block relationship, and boundary conditions for input to the EAGLE (Ref. 1) and GMBE (Refs. 2 and 3) codes. BCON can accept input in the form of strings or arrays of points generated from any surface geometry source. The user needs to prepare the surface geometry definition, the surface grids, and decide on the block topology. CAD tools or other geometry codes may be used for this purpose. BCON is written in C and uses NASA-Ames Research Center's PANEL Library for a graphical user interface on Silicon Graphics Inc.'s IRIS graphics workstations. The program is designed and developed in a highly modular structure which allows new features to be incorporated with minimal effort.

Chapter 3 of this volume, 'PROGRAM EXECUTION', provides basic information on how to run the program. The block topology and data specifications are described in Chapter 4: INPUT STRUCTURE. Two simple examples and one realistic example of a generic wing/body/strut/nacelle configuration from NASA-Langley are given in Chapter 5: EXAMPLES.

Each BCON function is described in detail in Chapter 6: FUNCTIONS.

Major functions provided in the program are :

1. Input;
2. Define block;
3. Define local index system;
4. Impose boundary conditions;
5. Write EAGLE run stream and input data deck;
6. Write GMBE block-block relationship and boundary conditions input files.

The program is delivered in the form of source code along with a build script (Makefile). BCON file formats are given in Appendix A. The process for building the BCON executable is described in Appendix B.

New users might first want to peruse Chapter 7: GENERAL RELEASE NOTES for an overview of the program capabilities.

3. PROGRAM EXECUTION

The source code, executable, the PANEL library, and the examples are stored in the following tree structure:

~/BCON
~/BCON/bcon: executable
~/BCON/examples
~/BCON/examples/case1: see example in Section 5.1
~/BCON/examples/case2: see example in Section 5.2
~/BCON/examples/case3: see example in Section 5.3
~/BCON/src_lib: BCON source codes
~/BCON/panel_lib: PANEL Library

To bring BCON up on an IRIS workstation, the user needs to move to a directory that has 'write' privileges (e.g. ~/BCON/examples/case1), and type '~/BCON/bcon'. This opens two MEX windows. These are: the 'Display Window' to provide the user with an interactive graphics environment; and the 'Main menu' window with pop-up/pull-down menu for the functions described in Chapter 6. All the usual IRIS window management functions are available in each BCON window.

The user can :

1. Load the input data file by picking the 'Input' option from the 'I/O' button on the 'Main menu';
2. Manipulate the graphics utility window 'Viewing' to translate, rotate, and scale the model;
3. Define blocks and the local coordinate index system through the 'Blocking' and 'Local index system' windows, respectively;
4. Impose boundary conditions by selecting appropriate options from the 'Bnd Cond' button on the 'Main menu';
5. Write EAGLE/GMBE input files by picking the 'Output' option from the 'I/O' button on the 'Main menu'; and
6. Exit the BCON system by hitting the 'Quit' button on the 'Main menu'.

Caution: The code does not provide journaling capability. That is, it does not record what the user has done before the results (i.e. EAGLE run stream and input data deck, block-to-block relationship, and boundary conditions files) are saved. The user is at risk of losing their efforts in the event of a system crash or an inadvertent exit from the code.

4. INPUT STRUCTURE

4.1 BLOCK TOPOLOGY STRUCTURE

The surface geometry definition, the block topology, and the edges and faces must be prepared before BCON can be used. It is recommended that the input data be furnished in the following sequence:

1. Loft (wing/body/strut/nacelle) surfaces individually, do surface-surface intersections, and ensure surface quality;
2. Design the block topology. Decide block boundaries, block shape, grid density, grid lines within the block, and across the block boundary;
3. Generate faces/sub-faces only on the configuration surfaces and the block faces where grid distribution preservation is required, and generate edges for the rest of the block faces.

Once the edge or face has been generated and written in the form of a string or array of points (the data structure is described in Section 4.2.1), BCON is used to glue all the blocks together and impose the appropriate boundary conditions on the model.

Neighboring blocks can be :

1. Multiple blocks connected to one block (figure 4.1);
2. Simply connected (figure 5.1); and
3. Partially connected (figure 5.2).

Each block can be formed by a combination of edges/sub-edges and faces/sub-faces. To simplify the data preparation and to reduce redundancy, unique edges and faces are required. This means that once a face or sub-face is defined, the edges or sub-edges surrounding it must not be redefined elsewhere. This is also true for edges and faces shared by neighboring blocks. In addition, edges and faces have free orientation. This means that the edge can be started at either end and that no specific orientation is required for the face columns and rows.

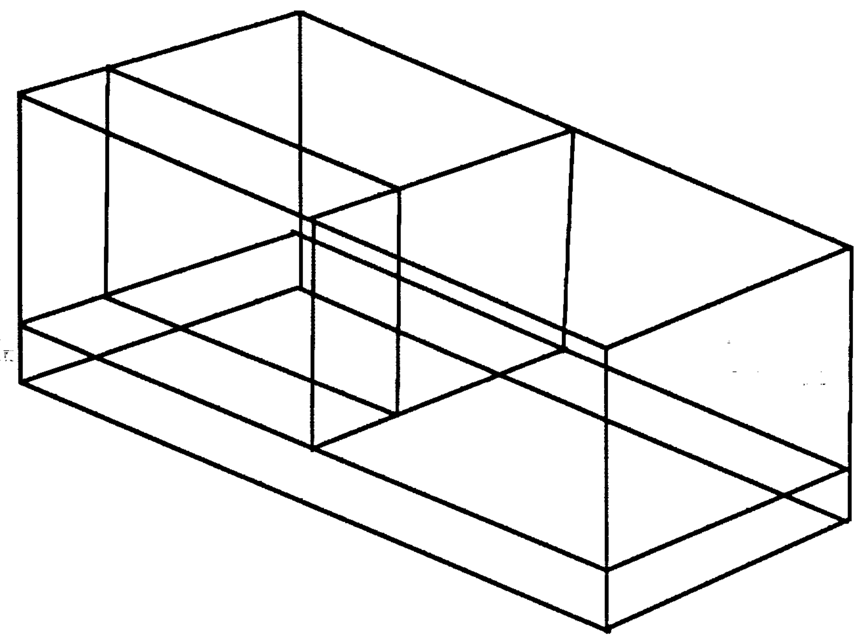

Figure 4.1 Multiple blocks connected to one block

4.2 DATA STRUCTURE

4.2.1 BCON Input File

The input file specifies the block definition as a combination of edges or faces which are represented by strings or arrays of points. For simplification, each block should be represented by as many edges as possible. Faces should be used only when it is necessary to preserve the original geometry or interface point distributions. Once a face is defined, the edges associated with that face need not be prepared. BCON will automatically identify the corresponding edge.

The generic name given to this file is 'agps.wfd'. The detailed format definition is given in Appendix A.1. Please refer to the directory ~/BCON/examples for sample files. The agps.wfd file for simply connected blocks (case 1) is listed in the Appendix C for reference.

4.2.2 EAGLE Run Stream File

This file format basically follows the EAGLE specification and can be produced by the BCON code.

The generic name given to this file is eagle.jcl. Please refer to the directory ~/BCON/examples for sample files.

4.2.3 EAGLE Input Data Deck File

EAGLE input data deck file specifies the block surface definition in the form of edges and faces, which are represented by strings/arrays of points, for each block throughout blocks. BCON produces a CRAY-YMP binary file which includes all block definitions in one single file. EAGLE uses this binary file to produce the volume grid.

The generic name for this file is 'block.bin'. Please refer to the directory ~/BCON/examples for sample files.

4.2.4 GMBE Block-to-Block Relationship Input File

This input file specifies the block-to-block relationships on the six faces of each block and provides the appropriate data communication between neighboring blocks for the GMBE calculation. Each face is defined with a unique record number and a set of running indices to serve as a communication tool between blocks. For each block, the local coordinates system (i.e. coordinate indices I, J, and K) follows the right-hand rule. GMBE, however, requires that the local coordinate index I be consistent, and along the flow direction, throughout blocks. This simplifies the algorithm and coding for the block-interface data communication. Six faces are defined in the following convention: Faces one and two are constant I-planes, and I = IMINR = 2 and I = IMAXR - 1 correspond to face one and face two, respectively; similarly, faces three and four are constant J-planes, and J = JMINR = 2 and J = JMAXR - 1 for face three and face four; faces five and six are constant K-planes, and K = KMINR = 2 and K = KMAXR - 1 for face five and face six. Note that the local coordinate indices (i.e. IMAXR, JMAXR, and KMAXR) in this file are equivalent to the actual block dimension (i.e. INUM, JNUM, and KNUM) plus two which represent one extra layer of overlapped points on each side of the block.

The generic name for this file is 'relo.dat'. The detailed format definition for this file is available in Appendix A.2. Please refer to the directory ~/BCON/examples for sample files.

4.2.5 GMBE Boundary Conditions Input File

This input file specifies the boundary conditions on the six faces and/or their sub-faces of each block and provides the proper imposition of boundary conditions, such as interface, solid surface, inlet, two types of exhaust, far field, for the GMBE calculation. Each face is defined as a patch with a set of running indices to impose one type of boundary condition. Six faces are defined in the following convention: Faces one and two are constant I-planes, and I = IMINB = 2 and I = IMAXB correspond to face one and face two, respectively; similarly, faces three and four are constant J-planes, and J = JMINB = 2 and J = JMAXB for face three and face four,; faces five and six are constant K-planes, and K = KMINB = 2 and K = KMAXB for face five and face six. Note that the local coordinate indices (i.e. IMAXB, JMAXB, and KMAXB) in this file are one unit more than the actual block dimensions (i.e. INUM, JNUM, and KNUM) because of an extra layer of overlapped points on the side of block with smaller index number.

The generic name for this file is 'bco.dat'. The detailed format definition for this file is available in Appendix A.3. Please refer to the directory ~/BCON/examples for sample files.

5. EXAMPLES

This chapter is a tutorial on how the program runs. The first two examples use simplified topology to illustrate program operations. The third example is a realistic airplane geometry. The procedures given here are merely for illustrative purposes. Detailed instructions on the various options are given in Chapter 6. However, <u>it is important to note that only the left-mouse button is used to activate all BCON menu choices</u>.

Section 5.1 presents a two-block model having simple edges and faces. Section 5.2 presents a model with two blocks that are partially connected. The geometry has a mixture of simple and complex edges and faces. A generic wing/body/strut/nacelle configuration from NASA-Langley is given in Section 5.3.

The following notations are adopted in Chapter 5: b#, f#, and e# represent block number, face and edge, respectively. The filenames convention referred throughout the manual, as well as their relevant processes, are described in the Glossary.

5.1 CASE 1 : TWO SIMPLY CONNECTED BLOCKS

Please refer to figure 5.1. In this example, two simple blocks are given. They are defined in agps.wfd as follows: five simple edges (e1, e2, e5, e6, & e9) and two simple faces (f1 & f3) for block 1; and five simple edges (e3, e4, e7, e8, & e10) and two simple faces (f1 & f2) for block 2. They share a simple face (f1). All ten edges and three faces are generated externally and saved in agps.wfd in directory ~/BCON/examples/case1. This file is also listed in Appendix C.

Section 5.1.1, Read Faces/Edges, describes the program loading sequence for the model. The block definition procedures are given in Section 5.1.2, Define the Blocks. Section 5.1.3, Define the Local Coordinate System, shows the user how the local index system is assigned to the first block and propagated to the second block. Section 5.1.4, Impose the Boundary Conditions, describes how the boundary conditions can be imposed on the model. The processes to generate input files for EAGLE and GMBE codes is explained in Sections 5.1.5 and 5.1.6, respectively.

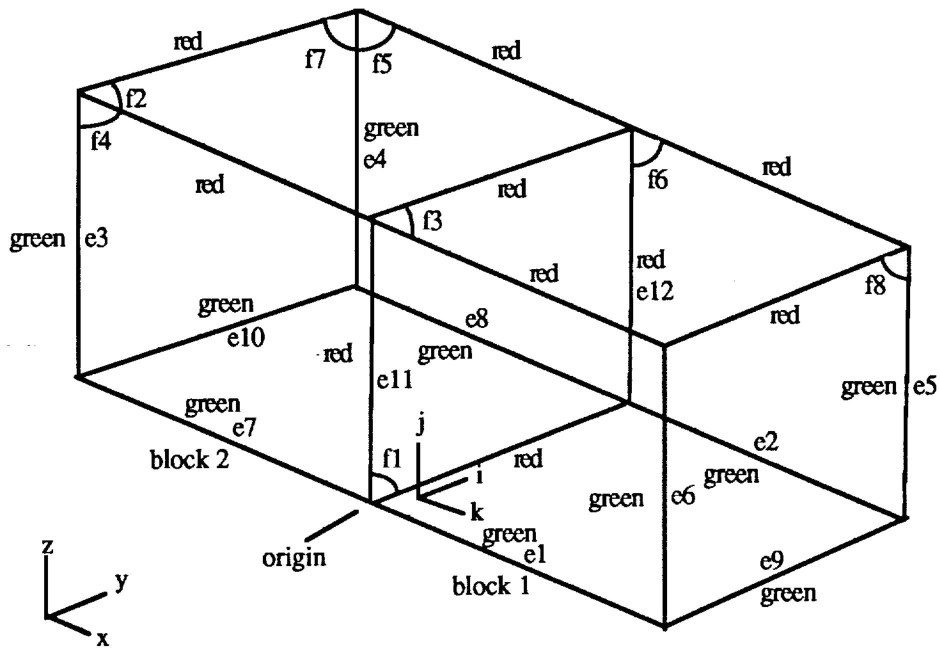

Figure 5.1 Two simply connected six-face blocks

5.1.1 Read Faces/Edges

Go to the directory ~/BCON/examples/case1. Type '../../bcon'.

The 'Main menu' will appear on the top right hand side of the screen, and the 'Display Window' will appear on the left.

1. Move the cursor to the 'I/O' button on the 'Main menu'. Push and hold the left mouse button down and move the cursor to 'Input' option. Then release the mouse button.
2. The 'Input/Output' window will be displayed. This is used for file selection.
3. Move the cursor to the filename 'agps.wfd' on the file list and click the left mouse button to select the file. Then move the cursor to the 'Select' option and click the left mouse button to activate the data loading action.
4. When the loading process is completed, the 'Input/Output' window will disappear and the model with two blocks will be displayed on the screen, three faces in red and ten edges in green. Description lines (Section 6.2) show up as dashed lines.
5. Windows for 'Viewing' and 'Blocking' appear.
6. Move the cursor to the slide button in the Y-translation slot on the 'Viewing' menu. Push and hold the left mouse button to move the model. Release it when the model is in the position you wish. Follow

similar steps to display the two simply connected blocks in the center of the screen.

5.1.2 Define the Blocks

Block one and block two are both simple blocks; i.e. each face of each block is represented by one simple face. Only two opposite faces need to be defined to form a block.

Block one (b1) can be defined in three steps:

1. Move the cursor to any description line on face f1. Click the left mouse button to select the face. Face f1 will be highlighted in light blue.
2. Move the cursor on top of edge e5. Click the left mouse button. Edge e5 will be highlighted in white.
3. Move the cursor on top of edge e6. Click the left mouse button. Face f8 will be formed and highlighted in light blue for an instant. Then block b1 will be formed and displayed in dark blue.

Block two (b2) can be defined in a similar way:

1. Move the cursor on top of e11, click the left mouse button to highlight e11 in white. Then click on e12 to highlight f1 in light blue.
   This step is equivalent to step 1 for block one. That means the user can either select the description line on a face (f1) or select any two edges (e.g. e11 & e12) of the face (f1) to define it.
2. Next highlight e3 in white. Then click on e4. Face f7 will be formed and highlighted in light blue for an instant before block b2 is formed and displayed in purple.

Move the cursor to the 'Done' button on 'Blocking' menu and click the left mouse button to conclude this process and remove the 'Blocking' menu from screen.

5.1.3 Define the Local Coordinate System

When the 'Blocking' menu disappears from the screen, an asterisk will be displayed on one corner of the model, at the same time the 'Local index system' window appears.

The local coordinate system for the model can be defined in the following steps:

1. Move the cursor to 'Select origin' button. Click the left mouse button to cycle the asterisk around the vertices of the block. Stop at the vertex

shown in the figure 5.1. Then click on the 'OK' button to confirm the action.

2. The direction of the local coordinate index I will be displayed by a red bar beginning at the origin selected in step 1. Use the 'Select i direction' button. Click the left mouse button to cycle the red bar through three directions around the origin. Stop at direction shown in the figure 5.1. Then click on the 'OK' button to confirm the action.
3. Next the direction of the local coordinate index J will be displayed by a dark blue bar. Move the cursor to 'Select j direction' button to cycle the dark blue bar through two directions around the origin. Stop at direction shown in the figure 5.1. Then confirm the action with the 'OK' button.
4. Move the cursor to the 'OK' button to accept the local coordinate system, glue the two blocks together and set up the block-to-block relationship.

The local coordinate indices, i, j, and k, will be displayed for each block. The process of defining the blocks and their relationship is concluded. The next step is to impose boundary conditions.

5.1.4 Impose the Boundary Conditions

Click on the 'Bnd Cond' button to activate the 'Boundary Conditions' window. The model will be displayed in the form of ten shrunken faces which includes the two input faces (f2 & f3) but not the interface (f1). The default BC type in the window is 'Solid Wall', represented in red.

Assign boundary condition to faces as follows:

1. i. Click on the 'Select Faces' button to begin the face selection.
   ii. Move the cursor to any edge of f7 and click the left mouse button to define this as a solid face. Face f7 will be displayed in red.
   iii. Move the cursor to the 'Done selecting' button and click the left mouse button to terminate the selection process.
   iv. Click on the 'Accept BC' button to confirm the selection and update the data structure. The red highlight of f7 dims. The select faces procedure may now be repeated.
2. Click on the 'Inlet' button and repeat step 1 to impose the inlet boundary condition on f4. This is displayed in blue.
3. Click on the 'Exhaust 1' button and repeat step 1 to impose a type 1 exhaust boundary condition on f5. This is displayed in green.
4. Similarly impose a type 2 exhaust boundary condition on f6. This is displayed in yellow.

5. Click on the 'Done' button to conclude this procedure. Faces that have not been assigned a boundary condition will default to far field conditions (Chapter 6).

This two simply connected blocks model has now been furnished with a block definition, a block-to-block relationship, and appropriate boundary conditions.

5.1.5 Write EAGLE Run Stream and Input Data_Deck

The EAGLE run stream file (eagle.jcl) and the input data file (block.bin in CRAY-YMP binary) are generated as follows:

1. Move the cursor to the 'I/O' button on the 'Main menu'. Push and hold the left mouse button down while moving the cursor to 'Output' option. Keep holding down the button until the 'EAGLE' option is highlighted in blue. Then release the mouse button.
2. The 'Input/Output' window will be displayed on the screen for options.
3. Move the cursor to 'Typein' button, click the left mouse button, key in 'eagle.jcl' for the EAGLE run stream file, and depress the return key on the keyboard or click on the left-mouse button.
4. After both the EAGLE input data file (block.bin) and the EAGLE run stream file (eagle.jcl) have been saved, the 'Input/Output' window will close.

5.1.6 Write GMBE Block-to-Block Relationship and Boundary Conditions Input Files

The GMBE block-to-block input file (relo.dat) and the boundary conditions input file (bco.dat) are generated as follows:

1. Move the cursor to the 'I/O' button on the 'Main menu'. Push and hold the left mouse button down. Move the cursor to 'Output' option. Keep holding down to 'Euler' option, and then release the mouse button.
2. After the GMBE input files relo.dat and bco.dat have been saved, control will be returned to 'Main menu'.

Now that the EAGLE run stream file, the EAGLE input data deck, the GMBE block-to-block relationship file and the GMBE boundary conditions file have been created, the user may select 'Quit' button on the 'Main menu' to exit from the program.

5.2 CASE 2 : BLOCKS WITH MULTIPLE SUB-FACES

Refer to figure 5.2. This is an example of two blocks with a few complex edges and faces. These are: a complex face (composed of f5, f8, f9, & f10) for block one; and a complex face (composed of f5, f6, f7, & f8) for block two. The blocks share two common sub-faces (f5 & f8). All seven edges and nine faces are saved in agps.wfd in directory ~/BCON/examples/case2.

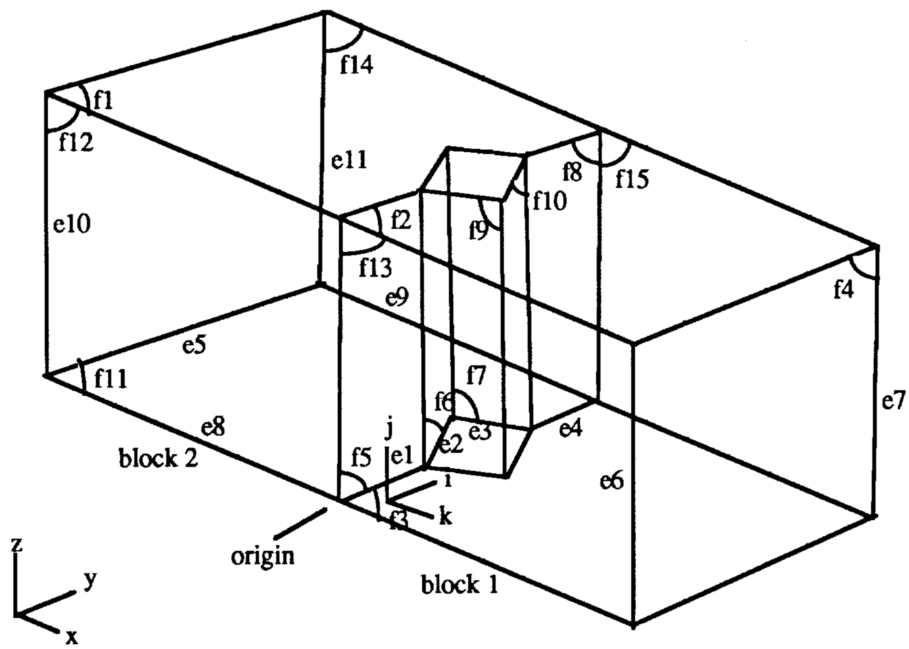

Figure 5.2 Two partially connected six-face blocks

5.2.1 Read Faces/Edges

Bring up BCON from the directory ~/BCON/examples/case2.

Use the I/O button to load the model. Move it to display the two partially connected blocks in the center of the screen.

5.2.2 Define the Blocks

One face of block one is composed of four sub-faces (f5, f9, f10, & f8). Its opposite face, formed by simple edges e6 and e7, needs to be defined for the block. All the four faces in between are represented either simple faces or they are faces formed by simple edges. The code should be able to pick up these

edges/faces automatically and form the block. For block two, one complex face (composed of f5, f6, f7, & f8) and one of its neighbor faces, formed by simple a edge e5 and a complex edge (composed of edges e1, e2, e3, & e4), must be defined. In order to form block two, a third face, opposite to the complex face or the face with one complex edge, needs to be defined. These blocks have two common two sub-faces, f5 & f8.

Block one (b1) can be defined in two steps:

1. First compose the complex face (f5, f8, f9, & f10). Click on the 'Complex' button under the 'Faces' option on the 'Blocking' window. This activates the defining process for a complex face. A new 'Done' button will appear alongside the 'Faces' 'Complex' button. Highlight f5, f8, f9, and f10 in white. Click the 'Done' button. A complex face across four sub-faces is formed. It is highlighted in light blue.
2. Use e6 and e7 to define f4 and thus form block b1. Block b1 is displayed in dark blue.

Block two (b2) can be defined in three steps:

1. Define the complex face, composed of sub-faces f5, f6, f7, & f8 as described in step 1 above. The next step is to define a complex edge.
2. Activate the 'Complex' button under the 'Edges' option in the 'Blocking' window. A new 'Done' button will appear next to it. Highlight e1, e2, e3, and e4 in white and then terminate the selection of sub-edges with the 'Done' button. There will be a complex edge, highlighted in white, across four sub-edges. Click on e5 to highlight face f11 in light blue.
3. Move the cursor to any description line (indicated by the red dots) on face f1. Click the left mouse button. Face f1 will be highlighted in light blue for an instant. Block b2 is then formed and displayed in purple.

Now that the two blocks have been defined, use the 'Done' button to quit the 'Blocking' menu.

5.2.3 Define the Local Coordinate System

Define the local coordinate system of this model according to figure 5.2. Follow the steps described in Section 5.1.3.

5.2.4 Impose the Boundary Conditions

Use the 'Bnd Cond' button to open the 'Boundary Conditions' window. The model will be displayed in the form of fourteen faces. These include three input faces (f1, f2 & f3), four input sub-faces (f6, f7, f9, & f10) but no inter sub-faces (f5 & f8).

The boundary conditions are imposed as follows:

1. Assign f6, f7, f9, and f10 solid wall boundary condition.
2. Impose inlet boundary condition to f12 and f13, type 1 exhaust boundary condition to f14, and type 2 exhaust boundary condition to f15.
3. Use the 'Done' button to close the 'Boundary Conditions' window. The purple and blue block definitions with their local coordinate system will re-appear in the 'Display' window.

The model is now furnished with block definitions, the block-to-block relationship, and appropriate boundary conditions.

5.2.5 Write EAGLE Run Stream and Input Data Deck

The EAGLE run stream file (i.e. eagle.jcl) and the input data file (i.e. block.bin in CRAY-YMP binary) can now be generated by following the steps given in Section 5.1.5.

5.2.6 Write GMBE Block-to-Block Relationship and Boundary Conditions Input Files

The GMBE block-to-block input file (i.e. relo.dat) and the boundary conditions input file (i.e. bco.dat) can be generated by following the steps given in Section 5.1.6.

Now that all relevant files have been written, one may use the 'Quit' button on the 'Main menu' to exit from the program.

5.3 CASE3 : WING/BODY/STRUT/NACELLE AIRPLANE CONFIGURATION

A generic wing/body/strut/nacelle airplane configuration from NASA-Langley will be demonstrated in this section. Please refer to figures 5.3.1.a through f and

5.3.2.a through z for the location of various faces and edges. In this example, twenty-six blocks some with complex edges and faces are given. The data file agps.wfd in the directory ~/BCON/examples/case3.

The reader may wish to refer to the procedures in Chapter 6 while working this example.

5.3.1 Read Faces/Edges

Bring up BCON from ~/BCON/examples/case3.

Load the model and display the twenty-six connected blocks in the center of the screen. Please refer to Figures 5.3.1.a through f for six different views of the model.

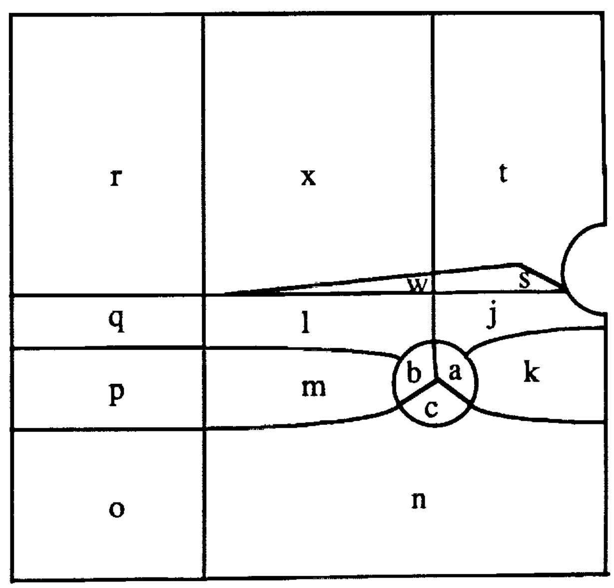

Figure 5.3.1.a Front View

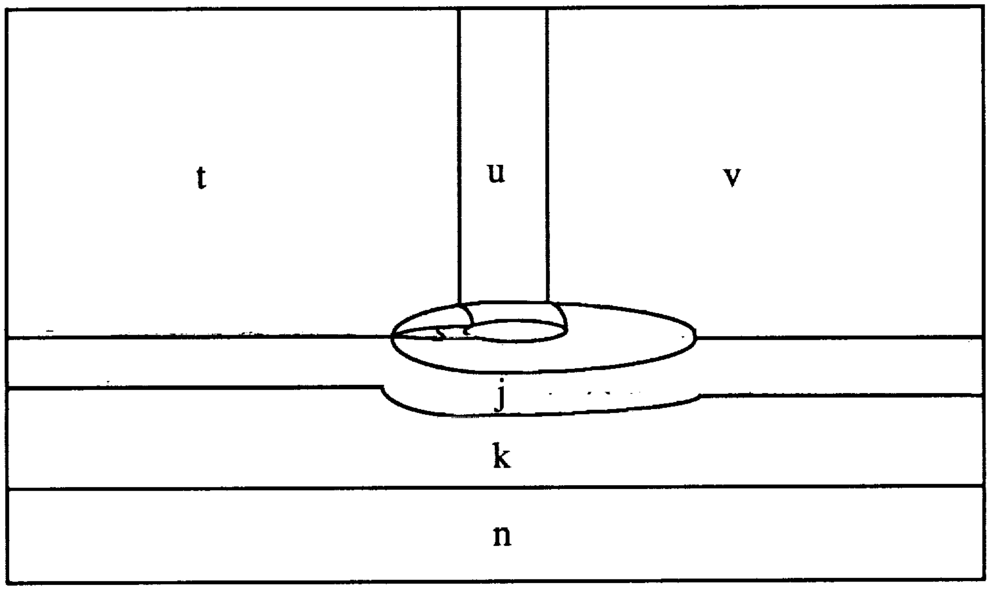

Figure 5.3.1.b Right View

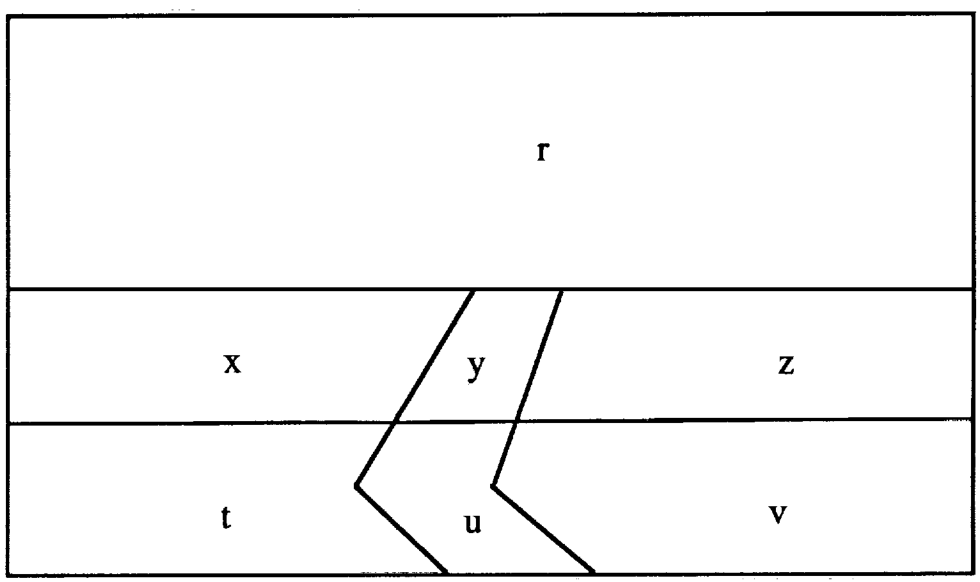

Figure 5.3.1.c Top View

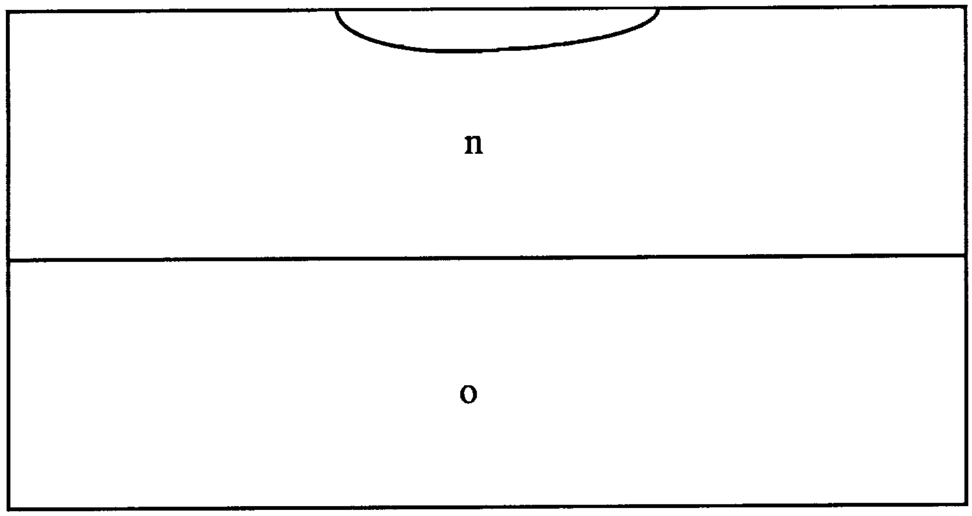

Figure 5.3.1.d Bottom View

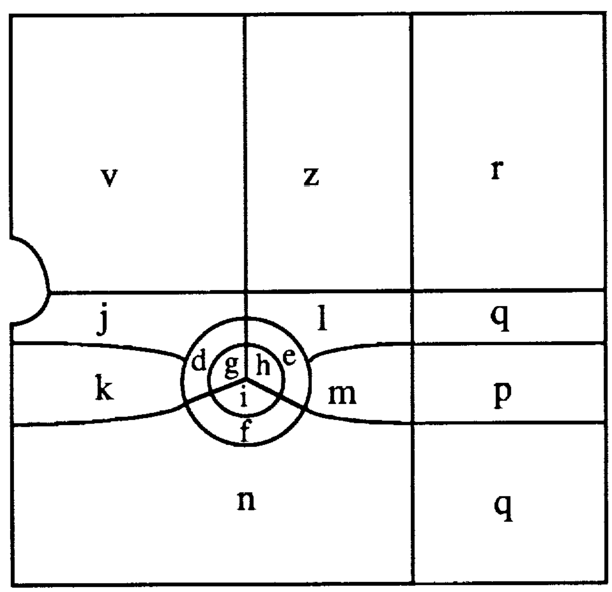

Figure 5.3.1.e Rear View

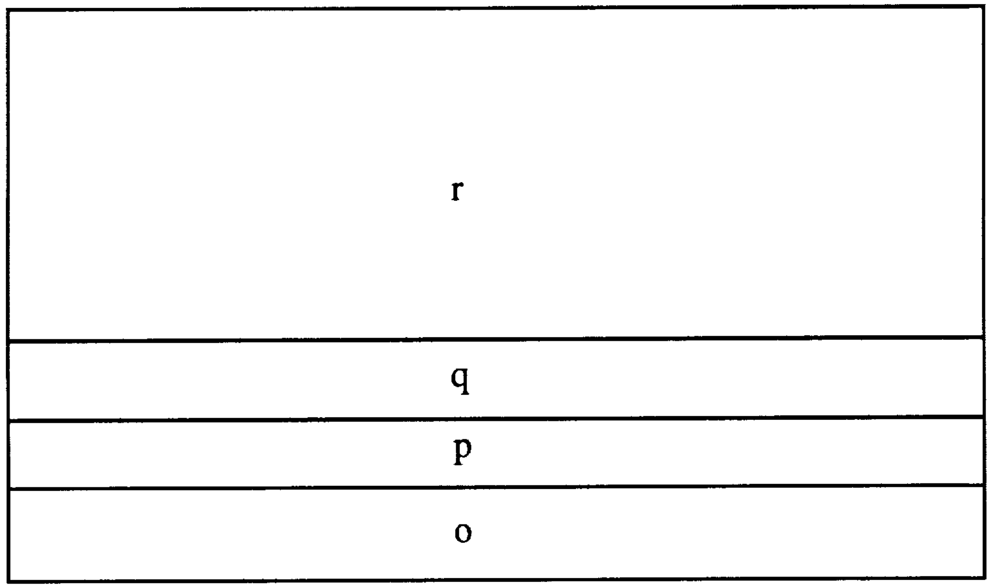

Figure 5.3.1.f Left View

5.3.2 Define the Blocks

Instead of showing the composite twenty-six blocks, figures 5.3.2.a through z present a sketch for each block to give user an idea of how each block is constructed. In these figures, complex edges are identified with either surface edges (e.g. The complex edge composed of e37 and e38 in figure 5.3.2.a) or hash marks (e.g. The complex edge composed of e1, e2, and e3 in figure 5.3.2.o). Note that all edges with discontinuities in slope are not necessarily complex edges (e.g. e69 in figure 5.3.2.t). Only the minimum required faces are defined in the figures, the remaining faces are simple faces. The blocks are defined as follows:

Block a	face one:	complex face composed of f37, f10, f43, & f44
	face two:	complex face composed of sub-face made of complex edge e33 & e34 and e36, and sub-face made of e47 and e50
	face three:	complex edge composed of e37 & e38 and complex edge composed of e36 & e50
	face four:	complex edge composed of e39 & e40 and complex edge composed of e41 & e47
	face five:	complex edge composed of e24 & e32 and edge e35
	face six:	f86 formed by selecting e112 and e113

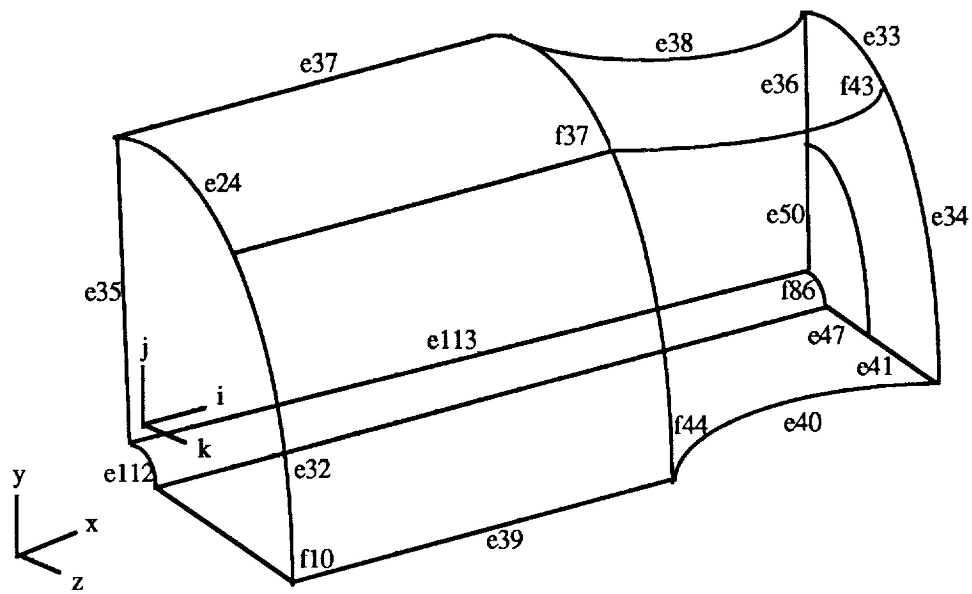

Figure 5.3.2.a Block a

Block b	face one:	complex face composed of f16, f22, f45, & f46
	face two:	complex face composed of sub-face made of complex edge e45 & e46 and e36, and sub-face made of e50 and e43
	face three:	complex edge composed of e37 & e38 and complex edge composed of e36 & e50
	face four:	complex edge composed of e48 & e49 and complex edge composed of e44 & e43
	face five:	complex edge composed of e16 & e42 and edge e35
	face six:	f87 formed by selecting e114 and e113

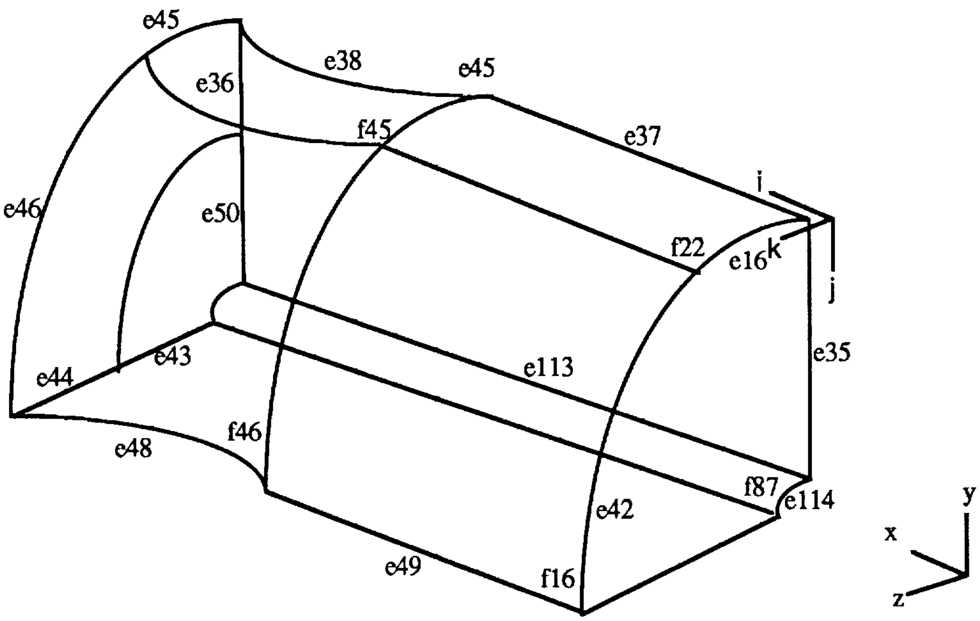

Figure 5.3.2.b Block b

Block c	face one:	complex face composed of f4 & f47
	face two:	complex edge composed of e39 & e40 and complex edge composed of e41 & e47
	face three:	complex edge composed of e48 & e49 and complex edge composed of e44 & e43
	face four:	complex face composed of sub-face made of e44 and e41, and sub-face made of e47 and e43

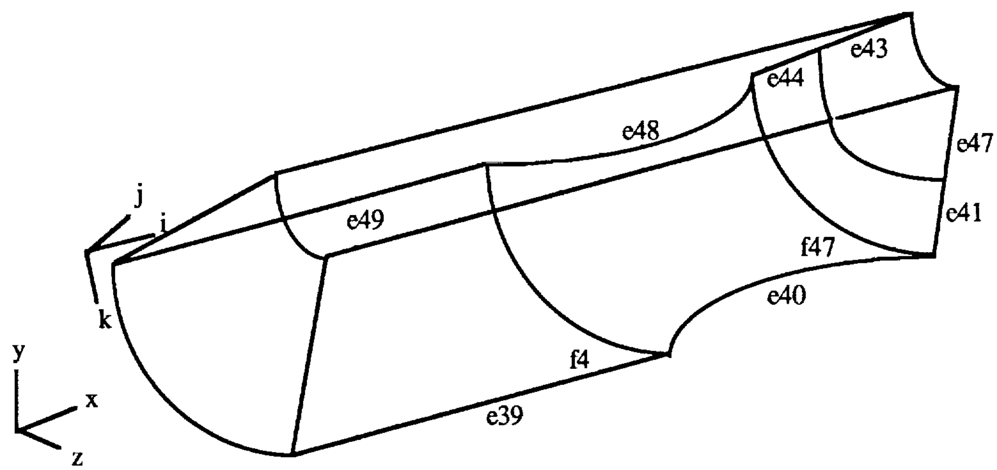

Figure 5.3.2.c Block c

Block d	face one:	complex face composed of f64 and sub-face made of complex edge composed of e51 & e52 and edge e53
	face two:	complex face composed of f60 & f61
	face three:	complex face composed of f39 & f63
	face four:	complex edge composed of e95 & e55, and edge e96

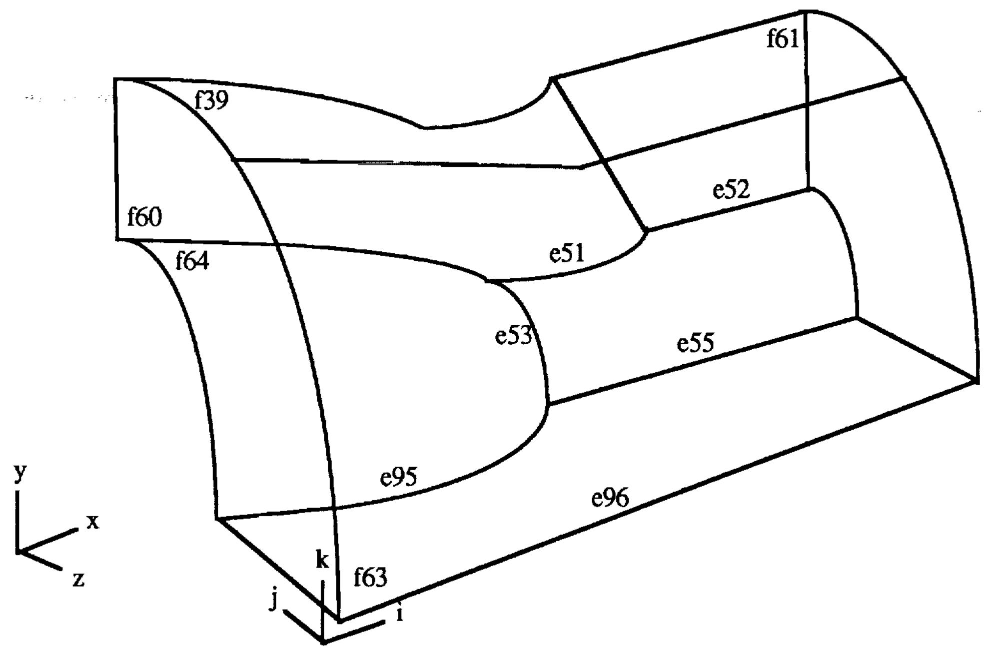

Figure 5.3.2.d Block d

Block e	face one:	complex face composed of f67 and sub-face made of complex edge composed of e56 & e52 and edge e57
	face two:	complex face composed of f68 & f61
	face three:	complex face composed of f24 & f66
	face four:	complex edge composed of e59 & e97, and edge e98

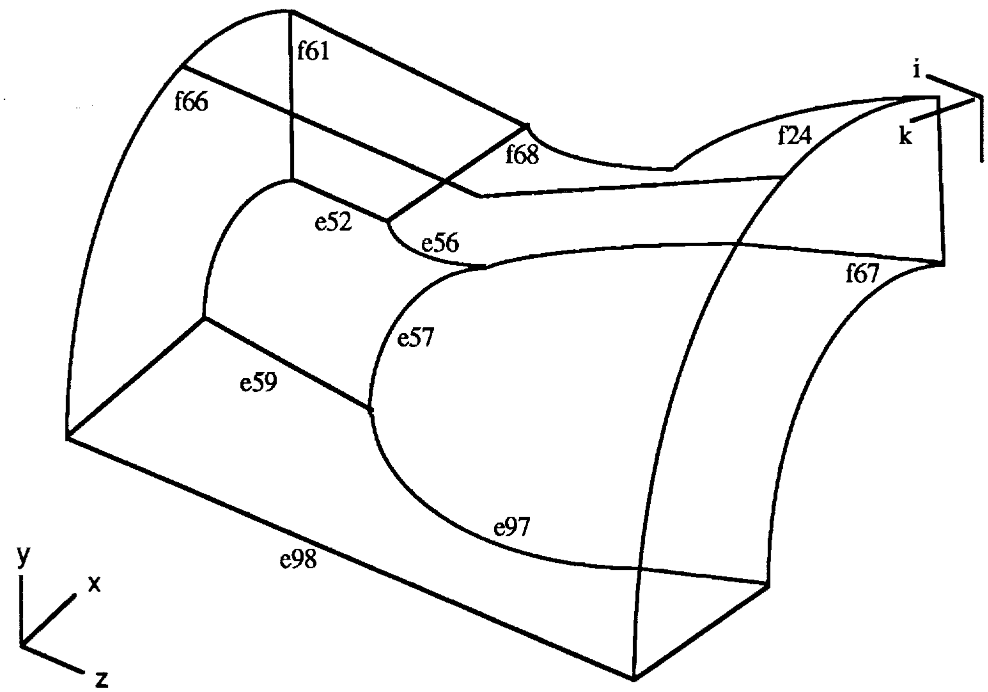

Figure 5.3.2.e Block e

Block f face one: complex face composed of f69 & f70
face two: complex edge composed of e95 & e55, and edge e96
face three: complex edge composed of e97 & e59, and edge e98

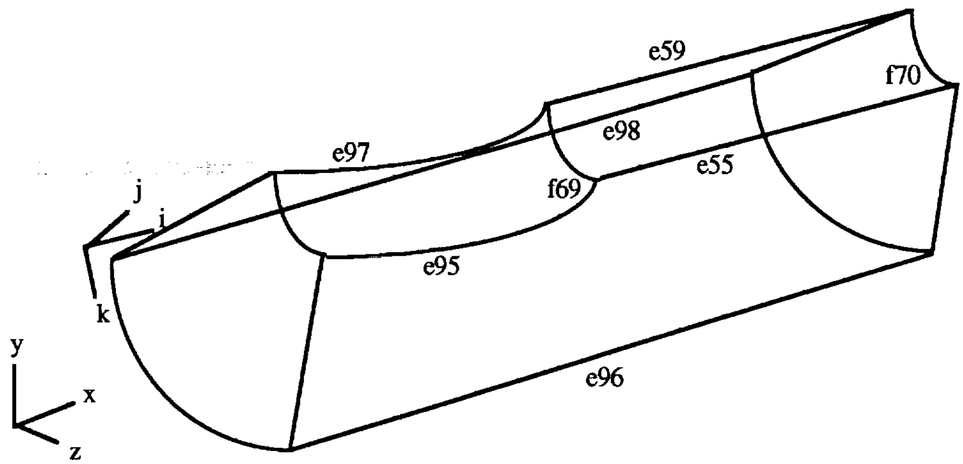

Figure 5.3.2.f Block f

Block g face one: complex face composed of f48 & f49
face two: complex edge composed of e51 & e52, and edge e53
face three: e54 and e55

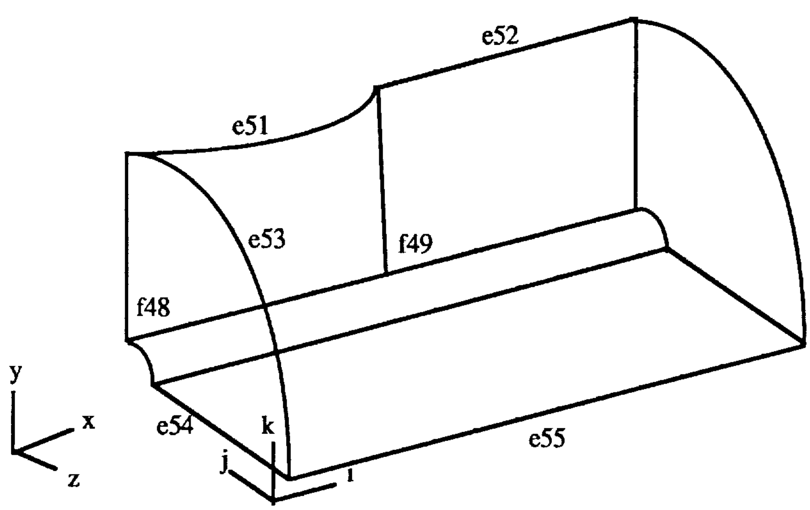

Figure 5.3.2.g Block g

Block h	face one:	complex face composed of f50 & f49
	face two:	complex edge composed of e56 & e52, and edge e57
	face three:	e58 and e59

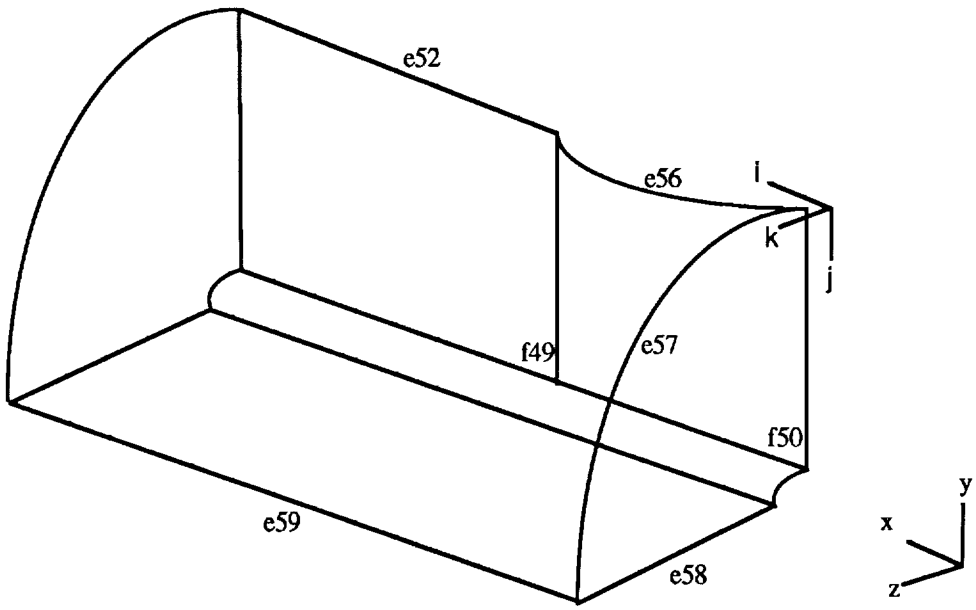

Figure 5.3.2.h Block h

Block i	face one:	e58 and e59
	face two:	e54 and e55

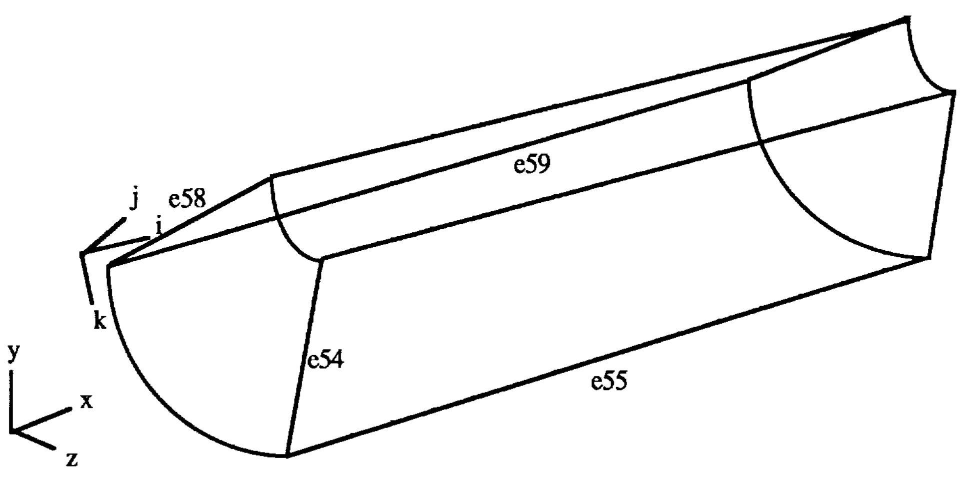

Figure 5.3.2.i Block i

Block j	face one:	complex face composed of f13, f14, f15, f37, f38, & f39
	face two:	complex face composed of f25, f40, & f27
	face three:	complex face composed of f34, f35, & f36
	face four:	complex face composed of f40, f41, & f42
	face five:	complex edge composed of e24 & e25 and edge e28
	face six:	complex edge composed of e26 & e27 and edge e29

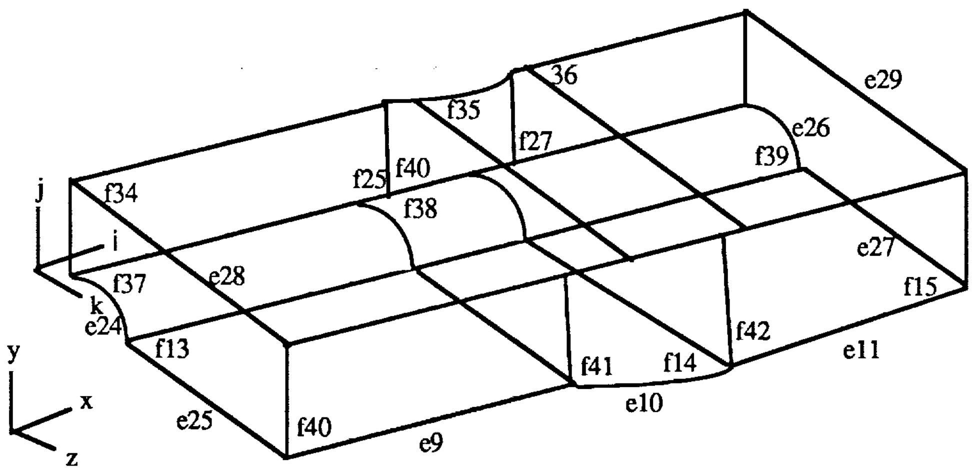

Figure 5.3.2.j Block j

Block k	face one:	complex face composed of f7, f8, & f9
	face two:	complex face composed of f10, f11, & f12
	face three:	complex face composed of f13, f14, & f15
	face four:	complex edge composed of e4, e5, & e6 and complex edge composed of e9, e10, & e11

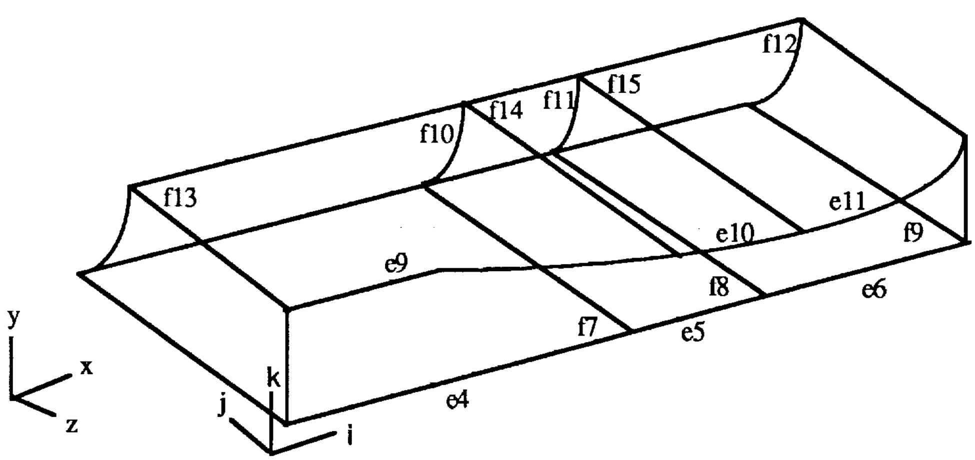

Figure 5.3.2.k Block k

Block 1	face one:	complex face composed of f19, f20, f21, f22, f23, & f24
	face two:	complex face composed of f25, f26, & f27
	face three:	complex face composed of f28, f29, f30, f31, f32, & f33
	face four:	complex edge composed of e15 & e16 and complex edge composed of e19 & e30
	face five:	complex edge composed of e17 & e18 and complex edge composed of e20 & e31
	face six:	complex edge composed of e12, e13, & e14 and complex edge composed of e21, e22, & e23

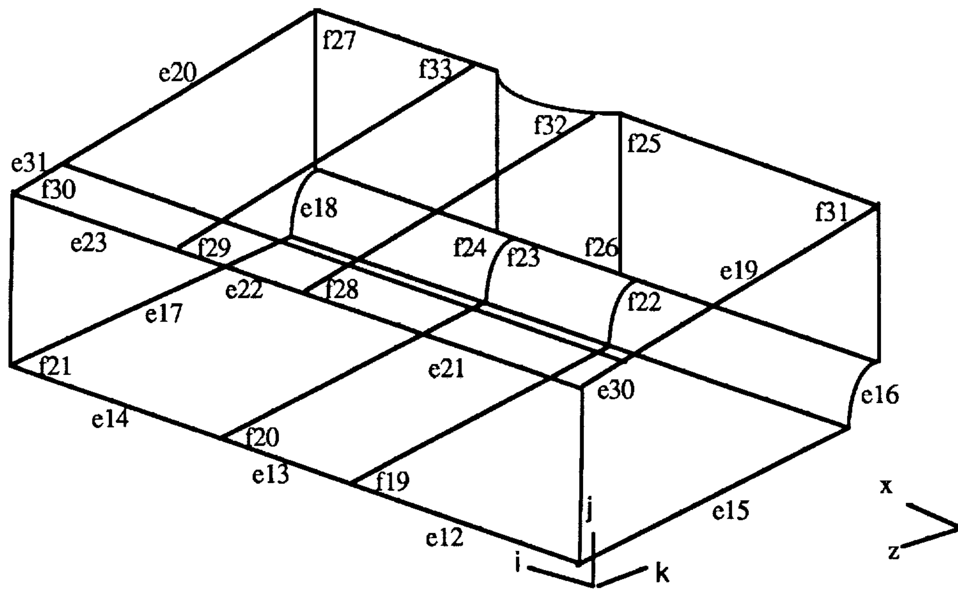

Figure 5.3.2.1 Block 1

Block m face one: complex face composed of f1, f2, & f3
face two: complex face composed of f16, f17, & f18
face three: complex face composed of f19, f20, & f21
face four: complex edge composed of e1, e2, & e3 and complex edge composed of e12, e13, & e14

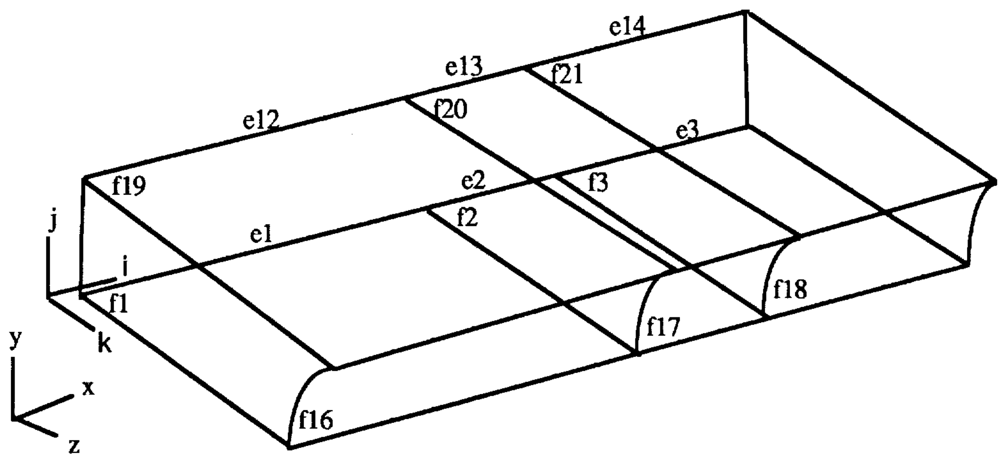

Figure 5.3.2.m Block m

Block n	face one:	complex face composed of f1, f2, f3, f4, f5, f6, f7, f8, & f9
	face two:	complex edge composed of e1, e2, & e3 and edge e7
	face three:	complex edge composed of e4, e5, & e6 and edge e8

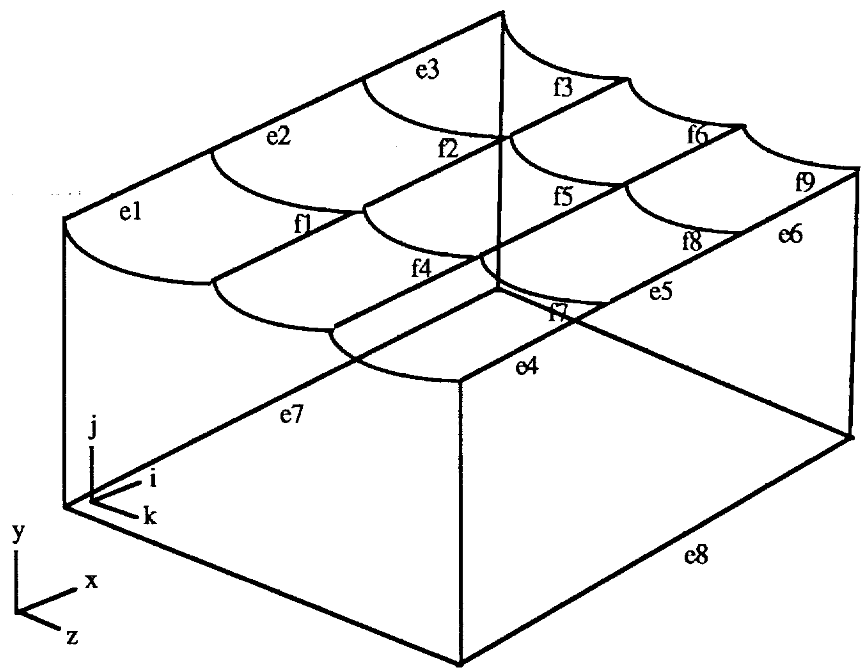

Figure 5.3.2.n Blcok n

Block o face one: complex edge composed of e1, e2, & e3, and edge e89
face two: complex edge composed of e1, e2, & e3, and edge e7
face three: e7 and e90

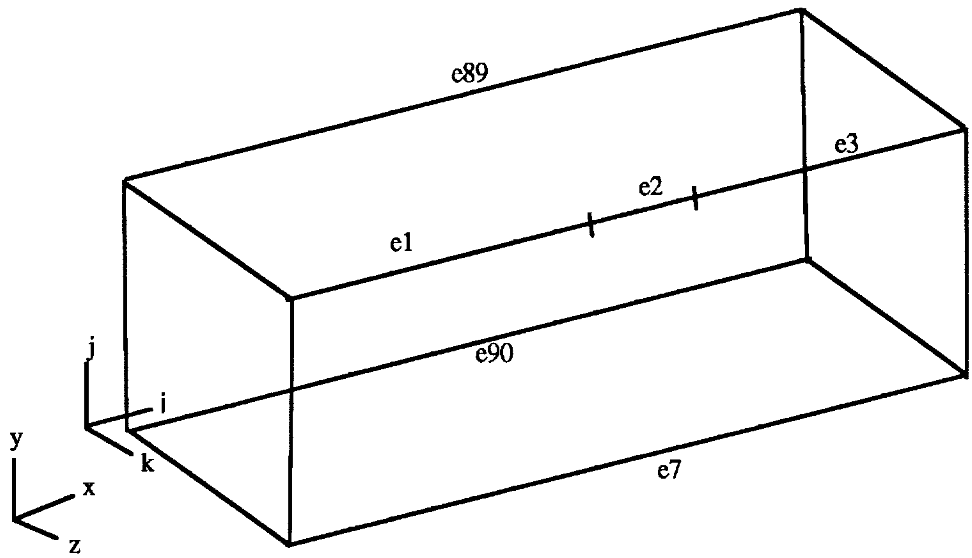

Figure 5.3.2.o Block o

Block p	face one:	complex edge composed of e1, e2, & e3, and complex edge composed of e12, e13, & e14
	face two:	complex edge composed of e1, e2, & e3, and edge e89
	face three:	complex edge composed of e12, e13, & e14, and edge e88

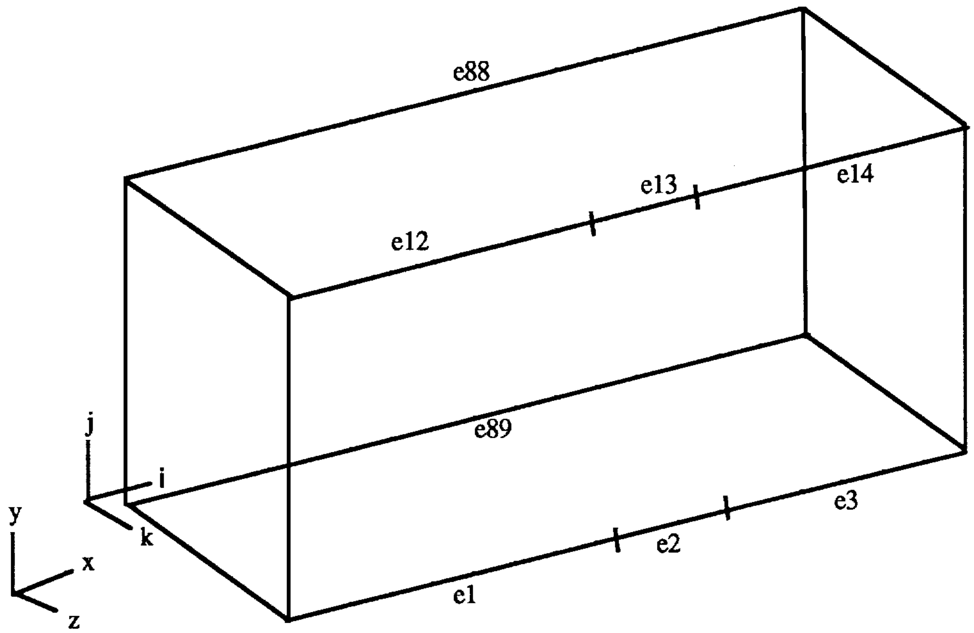

Figure 5.3.2.p Block p

Block q	face one:	complex face composed of sub-face made of e87 and e83, and sub-face made of complex edge composed of e22 & e23 and edge e83
	face two:	complex edge composed of e21, e22, & e23 and complex edge composed of e12, e13, & e14
	face three:	complex edge composed of e85 & e86 and edge e88
	face four:	complex edge composed of e12, e13, & e14 and edge e88

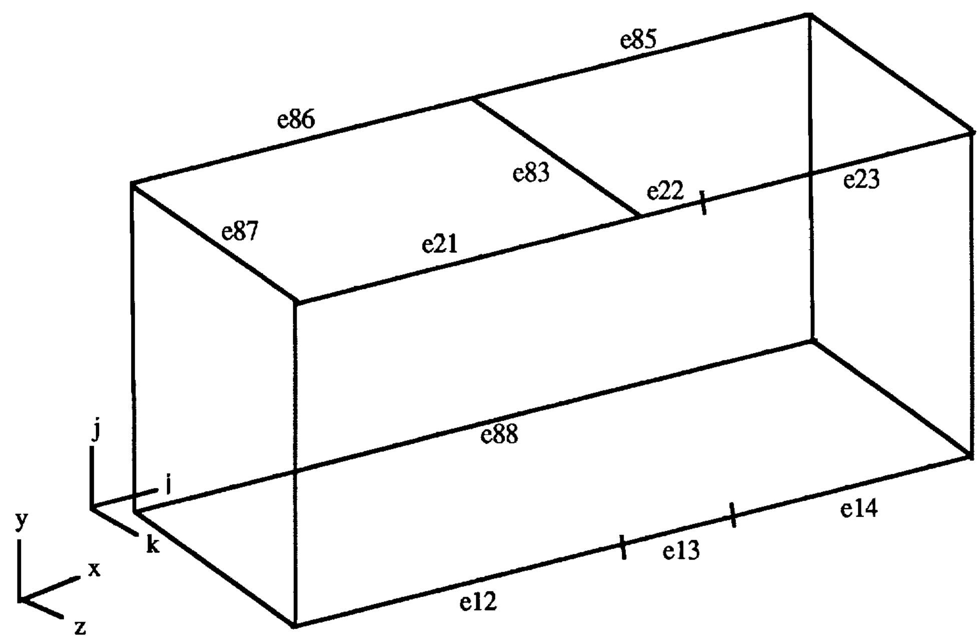

Figure 5.3.2.q Block q

Block r face one: complex face composed of sub-face made of e111 and e110, and sub-face made of complex edge composed of e80 & e81 and edge e84

face two: complex face composed of sub-face made of e111 and e79, and sub-face made of e84 and e79

face three: complex face composed of sub-face made of e110 and e64, sub-face made of e80 and edge e65, and sub-face made of e81 and edge e71

face four: complex face composed of sub-face made of e21 and e83, and sub-face made of complex edge composed of e22 & e23 and edge e83

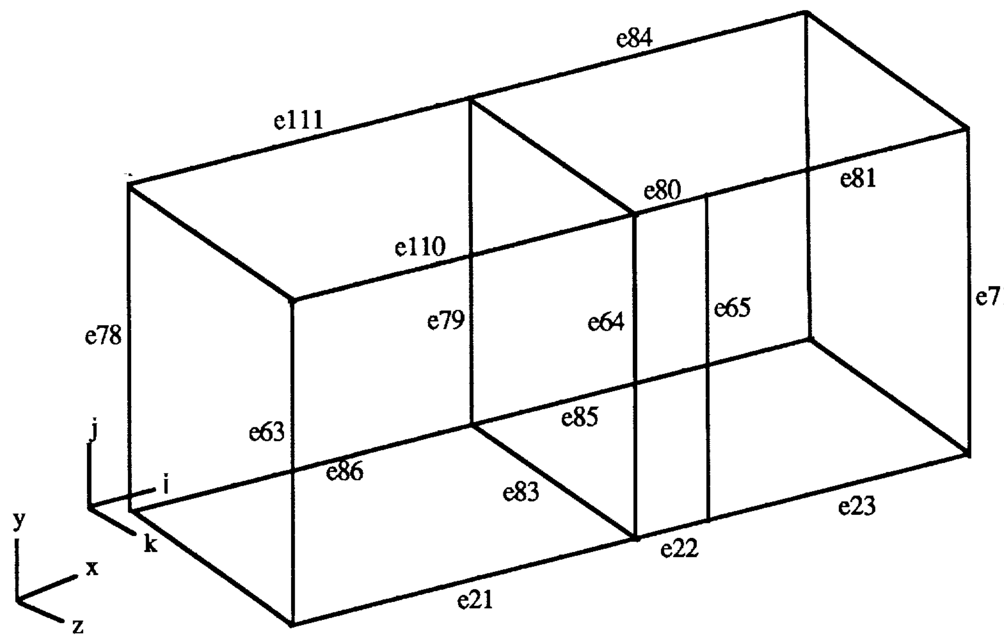

Figure 5.3.2.r Block r

Block s face one: e28 and e61
face two: f52

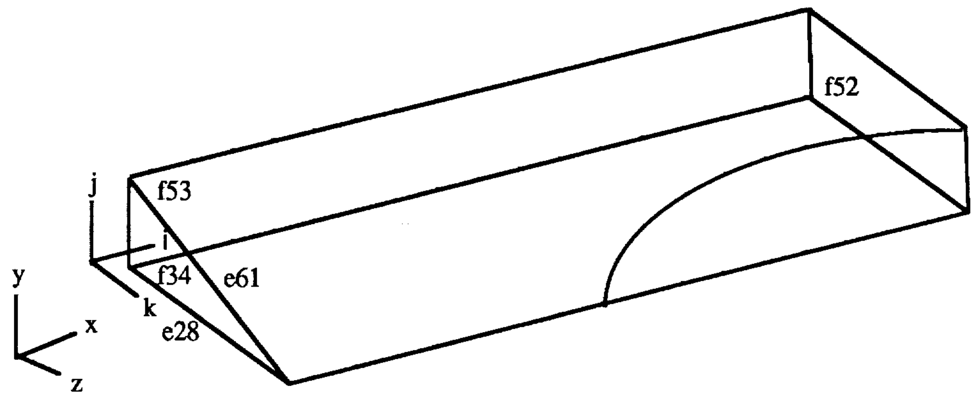

Figure 5.3.2.s Block s

Block t face one: e91 and e92
face two: f58

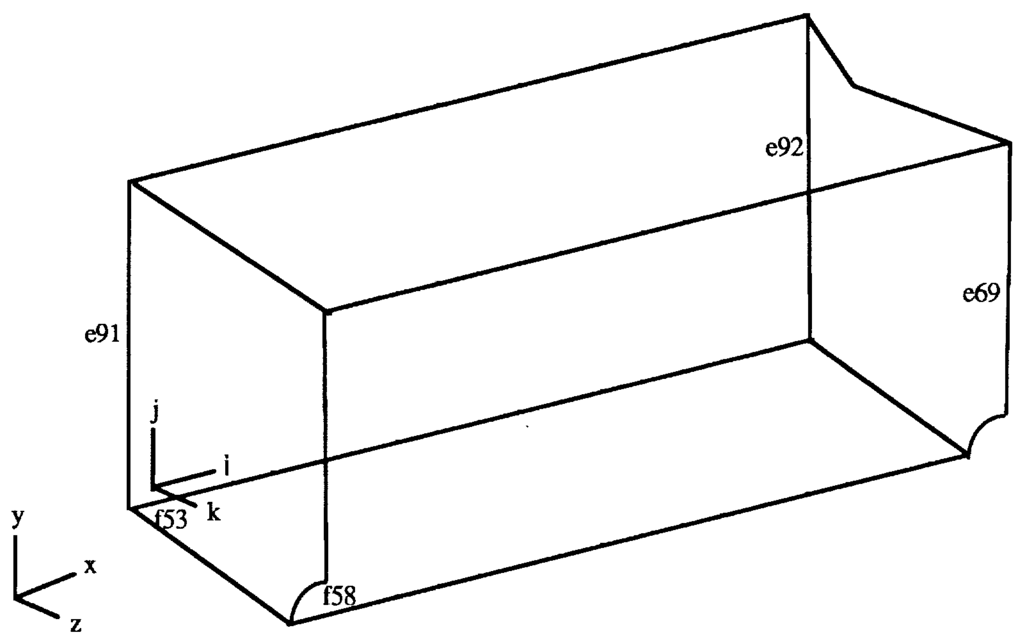

Figure 5.3.2.t Block t

Block u face one: f56
face two: e92 and e93

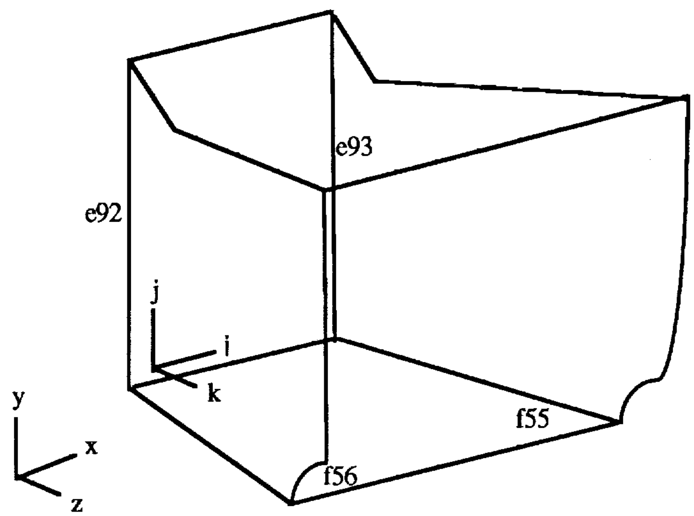

Figure 5.3.2.u Block u

Block v face one: e93 and e94
face two: f59

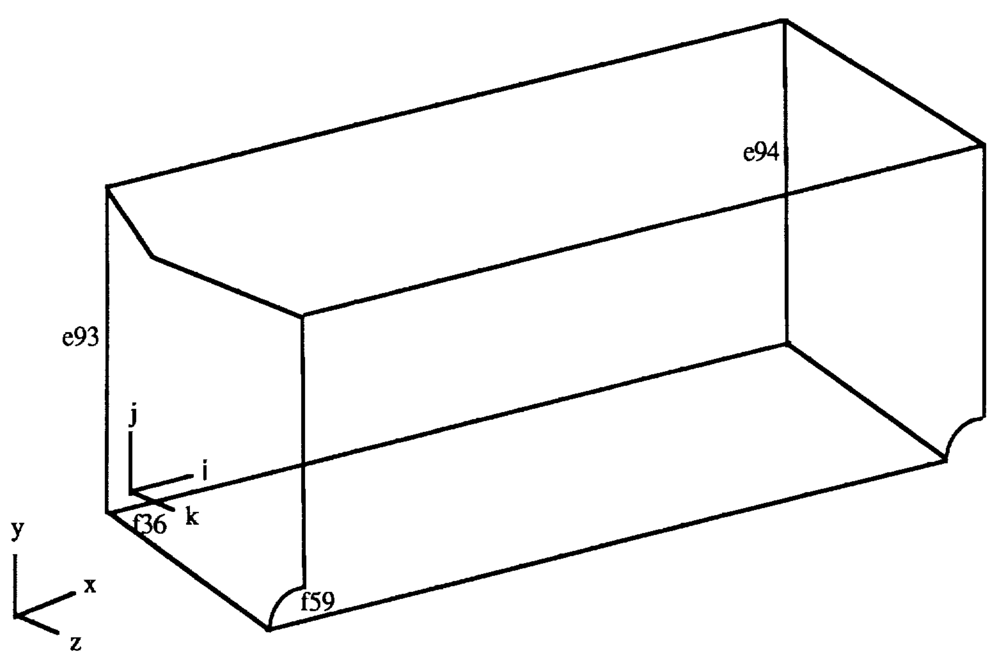

Figure 5.3.2.v Block v

Block w face one: e19 and e60
face two: f51

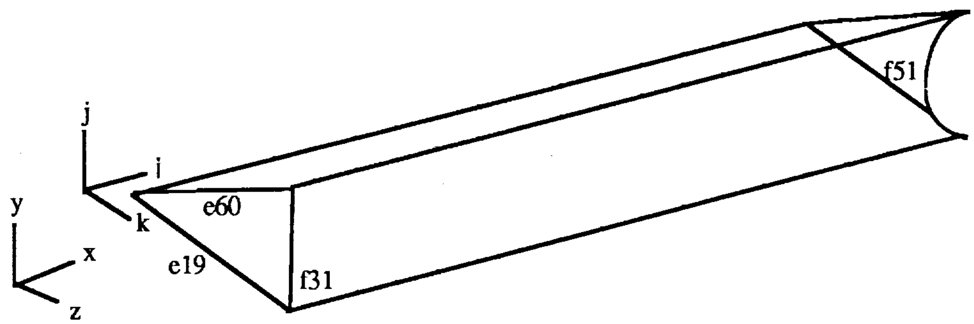

Figure 5.3.2.w Block w

Block x face one: complex face composed of f28 & f53
face two: complex edge composed of e30 & e60 and edge e63
face three: complex edge composed of e66 & e62 and edge e64

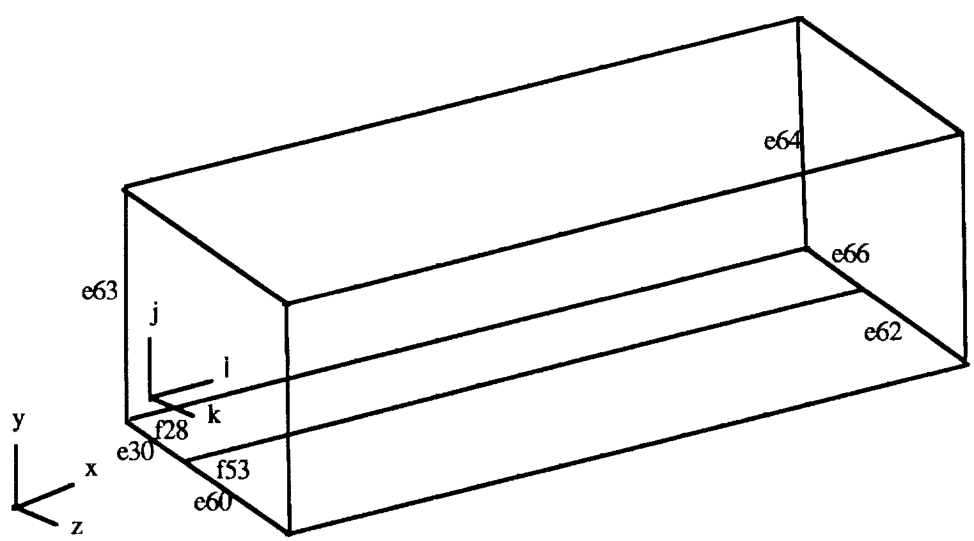

Figure 5.3.2.x Block x

Block y	face one:	complex face composed of f29 & f54
	face two:	complex edge composed of e66 & e62 and edge e64
	face three:	complex edge composed of e67 & e68 and edge e65

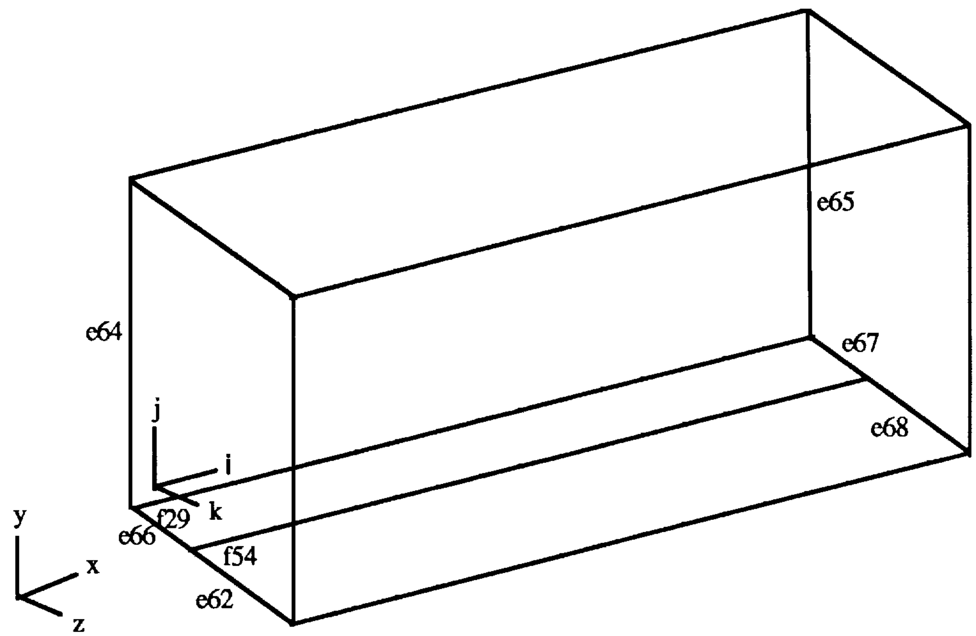

Figure 5.3.2.y Block y

Block z face one: complex face composed of f30 & f33
face two: complex edge composed of e67 & e68 and edge e65
face three: complex edge composed of e20 & e31 and edge e71

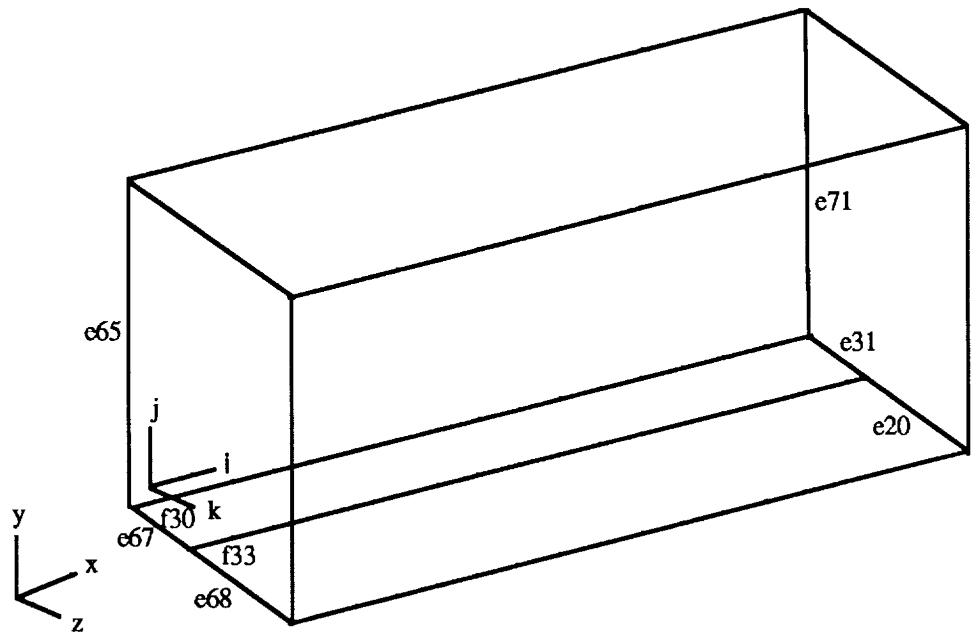

Figure 5.3.2.z Block z

5.3.3 Define the Local Coordinate System

Referring to figure 5.3.2.a, the local coordinate system of this model can be defined by following the steps described in Section 5.1.3.

5.3.4 Impose the Boundary Conditions

Follow the steps described in Section 5.1.4 to impose:

1. Solid wall boundary condition on the faces of the solids (i.e., wing, body, strut, and nacelle) and symmetry plane. For example, the wing faces are f32, f35, f51, f52, f54, and f55 in figures 5.3.2.l, 5.3.2.j, 5.3.2.w, 5.3.2.s, 5.3.2.y, and 5.3.2.u, respectively.
2. Inlet boundary conditions on the faces interfacing blocks a, b, and c with the nacelle.
3. Type 1 exhaust boundary condition on the faces interfacing the blocks d, e, and f with the nacelle. This is the fan exhaust.

4. Type 2 exhaust boundary conditions on the faces interfacing the blocks g, h, and i with the nacelle. This is the core exhaust.

Other faces will default to free stream conditions.

5.3.5 Write EAGLE Run Stream and Input Data Deck

The EAGLE run stream file (i.e. eagle.jcl) and the input data file (i.e. block.bin in CRAY-YMP binary) can now be written. Follow the steps in Section 5.1.5.

5.3.6 Write GMBE Block-to-Block Relationship and Boundary Conditions Input Files

The GMBE block-to-block input file (i.e. relo.dat) and the boundary conditions input file (i.e. bco.dat) can now be written. Follow the steps in Section 5.1.6.

Once all the relevant files have been saved, exit from BCON with the 'Quit' button.

6. FUNCTIONS

The main purpose of this chapter is to explain the operation of the functions available in BCON. It also contains brief explanations of the design background, wherever appropriate, to give the user a feel for how the program was developed. It is assumed that the user has some experience on IRIS multi-window and UNIX working environments. To develop a more clear understanding of the system, it is recommended that the user refers to examples given in Chapter 5 while reading this chapter.

First, the BCON code can be brought up by typing ~/BCON/bcon. When a new session begins, the internal data structure will be initialized automatically.

When BCON is up, the 'Main menu' will be shown on the top right of the screen, and the 'Display Window' takes up two-thirds of the screen on the left. The 'Display Window' provides the user with the interactive graphics working area that: displays the model that was defined with CAD tools or other geometry codes; takes the user's graphical input, and displays graphical responses to the user's input. The 'Main menu' window provides the user with a menu/mouse-driven user-interface; its lower levels of pop-up/pull-down menus are self-explanatory. It contains three major selections: 'I/O', 'Bnd Cond', and 'Quit'.

The 'I/O' button initiates a major process in BCON, it provides the functions to load a model from the designated directory, to define the blocks, to establish the local index systems for the blocks, and to output files for running EAGLE/GMBE codes. The design background and the operational procedures of these functions are described in detail in Sections 6.2, 6.3, 6.4, 6.5, and 6.6.

The 'Bnd Cond' button is initially de-activated when BCON is first brought up. It will automatically be activated once the processes of defining blocks and local index system are concluded. It automatically provides the functions to identify the block interfaces as far-field, solid, inlet and two types of exhaust boundaries. For GMBE, symmetry planes must be defined as solid boundaries. The design background and the operation procedures of these functions are described in detail in Section 6.5.

The 'Quit' button gives the user a normal exit from BCON. However, caution should be observed. Once the 'Quit' button is selected, the current session is concluded and the data structure will not be saved.

Section 6.1 is a summary of miscellaneous utilities, provided for quick reference.

6.1 UTILITIES

6.1.1 Mouse Key Definition

The left mouse key is used for the selection of all menu and screen graphical options during the BCON session.

6.1.2 Graphics Utility Window

The graphics utility window, 'Viewing', provides the user with more visualization control (i.e. zoom, translate, and rotate). The user may control the linear movement of the model horizontally and vertically in the 'Display Window' by moving the slide button in the X-translation and Y-translation slots, respectively; rotate the model around X-axis, Y-axis, and Z-axis by sliding the button in the X-rotation, Y-rotation, and Z-rotation slots, respectively; zoom the model by moving the sliding button in the scale slot; and reset all translation, rotation, and scaling parameters to bring the model back to its original view by clicking on the 'Reset' button. The change in the model seen in the 'Display Window' is linearly dependent on the location of the slide button in the slot. The farther a slide button is from the center of the slot, the larger is the change.

6.1.3 IRIS Mex Windows Environment

All the usual UNIX commands and IRIS mex window capabilities are available during the course of a BCON session. Windows can be moved with the middle-mouse button and can be altered (scaled, pushed, popped, etc.) with the right-mouse button. To activate an open BCON mex window, move the cursor control into it.

6.1.4 Quitting from the BCON Code

There are two ways to exit from the system: either, select the 'Quit' button on the 'Main menu' to exit the system normally; or, move the cursor control to the UNIX window from which BCON was initiated and type control C to abort the process.

6.2 INPUT

The input function is designed to load only one model per session. The input file is described in Section 4.2.1. The current code will not check for the possible duplication of data or the number of points on the related opposite edges. The user should ensure that the input data file contains unique data sets. There should be only one string or array of points per edge (sub-edge) or face (sub-face). In addition, the related opposite edges or faces must have the same number of points.

The procedure to get into the input function is as follows:

1. Select and hold the 'I/O' button on the 'Main menu'. Move the cursor to the 'Input' option on the pull down menu and release the mouse button. Note that the 'Output' option is unavailable at the start of a session.
2. A new window for input file selection, 'Input/Output', is displayed on the top of the 'Display Window'.
3. The user has options to change the working directory, to pick a file from the list of filenames, or to type in a filename with the appropriate path.
4. In order to change the working directory, select the 'Dir' option, type in the desired pathname and hit return.
5. To narrow down the filename search list, three options of file extensions are provided. Option 'all' lists all the files in the working directory. The default option lists files with the extension '.wfd' in the working directory. Option '?' allows user to specify the file extension. Once the filenames are listed, highlight the desired filename from the list by clicking on it. Then click the 'Select' button to load the model.
6. Option 'Typein' provides the user with the capability to key in the filename and its resident directory.

Once the model is loaded, the 'Input/Output' window will disappear from the screen. At the same time the model appears with its edges shown in green. The faces appear as solid red frames filled with red dashed lines which are drawn at every fourth grid line. The 'Viewing' and 'Blocking' windows are displayed. These allow the user to manipulate the model and define blocks.

6.3 HOW TO DEFINE A BLOCK

The 'Blocking' window provides the user with the utilities for defining blocks through interactive graphics. The user may screen-pick edges or sub-edges to define a face and then faces to define a block. At the end of process, all the defined blocks will automatically be "glued" together. Before going into the

details of the block defining procedure, the user should note that a simple edge of a block can be a sub-edge of a complex edge for a neighboring block. In the same manner, a simple face of a block can be a sub-face of a complex face for a neighboring block. Before a block can be defined, its edges and faces must be identified.

6.3.1 Define a Simple Edge

If one side of the face can be represented by a single edge that side is called a simple edge. This can take the form of an input edge (displayed in green) or one side of a single face (displayed in solid red). Click the 'Simple' button under the 'Edges' option in the 'Blocking' window. Move the cursor to the 'Display Window', and click on the desired edge in the model. The edge will be highlighted in white. That means one edge of the face is defined. The highlight will remain until a face of the block is formed. The user may click on the 'Reject' button, or press 'R' on the keyboard, to reject a selection. Repeated use of the 'Reject' button will step backwards through the process.

6.3.2 Define a Complex Edge

A complex edge refers to one side of a face that is composed of two or more sub-edges.

Define a complex edge as follows:

1. Chick on the 'Complex' button under the 'Edges' option in the 'Blocking' window. A 'Done' button will appear next to it.
2. Move the cursor to the 'Display Window'. Click on each sub-edge that forms the complex edge. These will be highlighted in white.
3. Click the 'Done' button when all sub-edges have been selected. A complex edge is now defined and will remain highlighted until a face is formed.

The user may click on the 'Reject' button or press 'R' on the keyboard to step backwards through the selection process. An additional reject will be needed to back out of the complex edge mode.

6.3.3 Define a Simple Face

When one face of the block can be represented by a single face, it is called a simple face. Click the 'Simple' button under the 'Faces' option in the 'Blocking'

window. Move the cursor to the 'Display Window', and define the desired face as follows:

1. If the face is a simple face, pick any one of the face description lines or any two edges of the face.
2. If the face is represented by four simple edges, pick any two edges. The code will automatically identify the other two simple edges.
3. If one side of face is represented by the complex edge, the complex edge must be defined following the procedure described in Section 6.3.2. Pick any one of the other three simple edges. The code will identify the remaining two simple edges that form the face.
4. If two or more sides of face are represented by the complex edges, each of the complex edges needs to be defined following the procedure described in Section 6.3.2. The code will identify any remaining simple edges of the face.

The face is highlighted in light blue, indicating that one face of the block is defined. The highlight will remain on until a block is formed. The rejection process for this function is the same as that described in Section 6.3.1.

6.3.4 Define a Complex Face

A complex face is composed of two or more sub-faces. A complex face can be defined either, by sub-faces, or, by a group of edges/sub-edges.

The procedure to define the complex face is as follows:

1. Click on the 'Complex' button under the 'Faces' option in the 'Blocking' window. A 'Done' button will appear next to it.
2. Move the cursor to the 'Display Window'. Select the desired sub-faces. These will be highlighted in white.
3. Click on the 'Done' button. The blue highlighted complex face will cover all the sub-faces.

The highlight will remain on until a block is formed. The rejection process for this function is the same as that described in Section 6.3.1. The program exits out of the complex face mode once the last sub-face is rejected.

6.3.5 Form the Block

A block must have six faces. It can be defined by one of the following methods:

1. If the block is represented by six simple faces, define any two opposite faces.
2. If the block contains complex edges or complex faces, first define all the complex faces, by following the methods described in Sections 6.3.2, 6.3.3, or 6.3.4. If any two of them are opposite faces, the code will automatically identify any remaining simple faces. Otherwise, the user needs to define the additional faces too.

Once a block is defined, BCON will assign a color to identify its edges. To conclude this blocking process, click on the 'Done' button that appears on the lower left corner of the 'Blocking' window. The user must ensure that all the blocks defined in this process are well-connected (i.e. no blocks in the model should be left unattached from their neighbors), before exiting from the 'Blocking' menu.

6.4 HOW TO DEFINE THE LOCAL COORDINATE SYSTEM

When the blocking process is concluded, the 'Blocking' window is removed from the screen and replaced by the 'Local index system' window. At this time, all the blocks are glued together and await the definition of local index system. The 'Local index system' window allows the user to interactively define the local indices i, j, and k for the first block. The code will propagate these indices throughout all the blocks defined in the blocking process. The block-to-block relationship prepared for EAGLE and GBME is based on the local index systems for the blocks defined in this section.

The procedure to define the local index system is as follows:

1. Only, 'Select Origin', button is activated when the 'Local index system' window is first brought up. This option gives user a freedom to choose one of the eight vertices as the origin (indicated by a white asterisk) of the local index system in the first block. The user can easily cycle the asterisk around the eight vertices via the '>' button. The choice is confirmed with the 'OK' button. This then de-activates the 'Select i direction' button.
2. Use the 'Select i direction' button. Cycle through and choose one of the three directions as the i-direction (shown as a red arrow).
3. The third option, 'Selection j direction', is activated, once the i-direction is set. Cycle through to choose one of the two remaining directions as the j-direction (shown as a dark blue arrow). At this time, the user has the choice of selecting either a right- or a left-handed local index system. Once the 'OK' button is clicked, this option becomes de-

activated and the local index system for the first block is displayed at the origin that was specified in step one.
4. The user has a final chance to change the origin by selecting the 'Redo' button to reject the local index system and start the process over. Otherwise select the 'OK' button to accept the defined local index system. The code will compute and display the local index systems for the remaining blocks.

This concludes the block definition and the establishment of block-to-block relationships. The next step is to impose boundary conditions. Note, however, that the user need not specify the boundary conditions before outputting the EAGLE files.

6.5 HOW TO IMPOSE BOUNDARY CONDITIONS

Clicking the 'Bnd Cond' button on the 'Main menu' will bring up the 'Boundary Conditions' window for imposing various boundary conditions; i.e. solid wall, inlet, and two types of exhaust. The model will be displayed in the form of shrunken faces/sub-faces, in white, with all the interfaces removed. The faces/sub-faces, that are not assigned any one of the above boundary conditions, will default to far field boundaries.

The operation procedure to impose boundary conditions is as follows:

1. Select a boundary condition from those listed on the right side of the window.
2. Click 'Select Faces' to activate the selection process (i.e. both options 'Reject Last Face' and 'Done Selecting' are activated and all the faces/sub-faces displayed on screen become selectable). Once a face is selected, the corresponding color (red for solid wall or symmetry plane, blue for inlet, green for exhaust 1, and yellow for exhaust 2) will be superimposed on it.
3. The user has the option of rejecting the selected face(s) by clicking on the 'Reject Last Face' button, or by clicking the 'Done Selecting' button which activates another two options, 'Accept BC' and 'Reject BC'.
4. The 'Reject BC' option gives the user a chance to reject all faces with the boundary condition and start over again from step one.
5. The 'Accept BC' option confirms the selection. The boundary conditions returns the window to its original state. The user may choose to impose other boundary conditions or conclude the process via the 'Done' button.

6.6 OUTPUT

The output function is designed to create the EAGLE run-stream and input data files and the GMBE block-to-block and boundary conditions input files. These output files are described in Sections 4.2.2, 4.2.3, 4.2.4, and 4.2.5. The user is prompted for the name of the EAGLE run stream file. However, the EAGLE input file, GMBE block-to-block input file, and boundary conditions input file are named by the system. These are block.bin, relo.dat, and bco.dat, respectively. Please refer to the Glossary for descriptions of these files.

6.6.1 Output for EAGLE Code

The procedure to get EAGLE output files is as follows:

1. Select and hold the 'I/O' button on the 'Main menu'. Move the cursor to the 'Output' button on the pull-down menu. This spawns another pull-down menu now with an 'EAGLE' button. Move the cursor to the 'EAGLE' button and release the mouse button.
2. This opens an 'Input/Output' window for typing in the output file name.
3. The user has the option to change the working directory.
4. Select the 'Typein' option and type in the filename for the EAGLE run stream file.

After the EAGLE input files (i.e. EAGLE run stream file and block.bin) have been saved, the 'Input/Output' window will disappear from the screen. The code is now ready for generating GMBE input files.

6.6.2 Output for GMBE code

Select and hold the 'I/O' button on the 'Main menu' and move the cursor to the 'Output' option on the pull-down menu. Then release the mouse button on the 'Euler' option of the next pull-down menu. GMBE input files, relo.dat and bco.dat, will be created and saved.

The user may now exit from BCON.

7. GENERAL RELEASE NOTES

Every attempt has been made to make this software as functional as possible. The user, however, should be aware of the following known limitations and problems, as well as the suggested work-arounds.

7.1 ON-LINE HELP

The graphical user interface menus were designed to be a self-explanatory. Therefore, an on-line help menu is not available. If the code does not respond in the expected manner (i.e. status/highlights/color), please refer to Chapter 6 and this Chapter for more information.

7.2 CONSTRUCTION OF COMPLEX FACE

Before the 'Done' button is selected during the complex face construction, described in Section 6.3.4, the user should ensure that all the selected sub-faces are well chosen and properly represent the face of the block. Otherwise, the system may crash if any sub-face has not been correctly connected with its neighbors.

7.3 REJECT FOR COMPLEX FACE

As described in Sections 6.3.2 and 6.3.4, the reject process for a complex face does not work as it does for a complex edge. In particular, the menu may not be able to return one step back when both complex edge and complex face are selected. The reject process might have to be repeated a few times before the menu returns to normal. One may then proceed as desired.

7.4 CONSTRUCTING NON-SIX-FACE BLOCK

The current version of BCON constructs six-face blocks. It, however, does not support either the blocks with degenerated faces (i.e. a five-face wedge and circular cylinder) or wrap-around six-face blocks (i.e. annular cylinder). It is recommended that the cylinder (both solid and annular) be split into three blocks. Make the wedge into a six-face block by adding a small sixth face for the degenerated face and impose the solid boundary conditions on the this new face. For example, see the three blocks in figures 4.3.2.a, 4.3.2.b, and 4.3.2.c that compose the fan inlet.

7.5 RE-PLAYABLE FUNCTION

The code supports neither journaling nor play-back functions. Whenever a new session starts, the data structure is initialized automatically.

7.6 AUTOMATED BLOCK TOPOLOGY

The block layout topology has to be done manually. CAD tools or other geometry codes might be helpful in this regard. BCON does not support this function.

7.7 DUPLICATED EDGE/FACE

Currently, the code does not provide a function to check for the duplication of edges and faces. The user must ensure that the input data file does not contain any duplicate edges or faces before loading it into BCON.

7.8 NUMBER OF POINTS ON THE RELATED EDGES

Since the code does not check the number of points on the opposite edges, it is recommended that the user should ensure the number of points on related edges are consistent.

7.9 INCREMENTS IN THE GRAPHICS UTILITY WINDOW

The graphics utility window, 'Viewing', described in Section 6.1.2, provides the user with visualization control. The current increment settings for the model translation, rotation, and scaling are linear.

7.10 RECOMMENDED FUTURE ENHANCEMENTS

1. Validate the number of points on the opposite edges/faces.

2. Define the complex face made by three or more sub-faces of which one sub-face shares edges with others. The current algorithm allows the complex

face shown in figure 7.10.2.a but not one in figure 7.10.2.b. Improve this to take the complex face shown as figure 7.10.2.b.

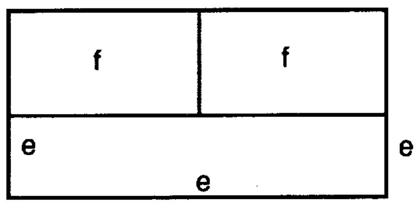

Figure 7.10.2.a Current algorithm allows this

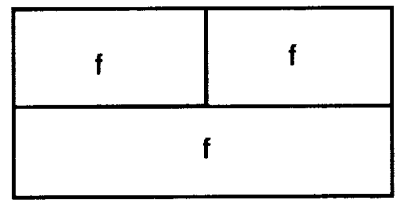

Figure 7.10.2.b Current algorithm does not allow this

3. Include a function to delete a defined block.

4. Include a user-controllable viewing distance for perspective view.

5. Provide an option to choose an isometric view.

6. Develop better visualization control of rotation, translation, and scaling.

7. Provide a 'working' signal to show work in progress.

8. Provide an option to specify all output file names and locations.

9. Include a process to define circular or annular cylinders.

10. Include a process to define five-face wedge blocks.

11. Make the reject process for complex faces consistent with the reject process for complex edges.

12. Enable the user to de-activate/de-hilite objects when they are rejected, as well as reset the option to restore selectability.

13. Provide an option for binary file input to BCON.

14. Provide play-back and journaling capability.

15. Allow separate input for each block or group of blocks (instead of as one large file).

16. Provide a bridge to the reject mode or to a do-nothing status if the user inadvertently press the done button before completing the composition of a complex face.

17. The current algorithm for picking faces prefers the selection of face description lines. Improve this to include edges.

8. REFERENCES

1. Lijewski, E., and Cipolla, J., "Program EAGLE User's Manual," USAF Armament Laboratory Technical Report, AFATL-TR-88-117, Eglin AFB, September 1988.

2. Chen, H.C., Su, T. Y., and Kao, T. J., "A General Multiblock Euler Code for Propulsion Integration, Volume I: Theory Document," NASA CR-187484, May, 1991.

3. Chen, H.C., "A General Multiblock Euler Code for Propulsion Integration, Volume III: User Guide for the Euler Code," NASA CR-187484, May, 1991.

GLOSSARY

The following notational conventions are used in this document.

simple edge	A string of points representing an edge of a block.
sub-edge	A string of points representing a partial edge.
complex edge	An edge of a block composed of two or more sub-edges.
simple face	An array of points representing a face of a block.
sub-face	An array of points representing a partial face.
complex-face	A face of a block composed of two or more sub-faces.
EAGLE	3D volume grid generator in the EAGLE program (Eglin Arbitrary Geometry ImpLicit Euler) which is jointly developed by Eglin Air Force Base and Mississippi State University.
GMBE	Boeing - developed General MultiBlock Euler Solver.
PANEL Library	NASA-Ames - developed graphical user interface toolkit for writing applications for the Silicon Graphics IRIS workstation family.
PLOT3D	NASA-Ames - developed computer graphics program to visualize the grids and solutions of computational fluid dynamics.

The glossary also includes brief descriptions of the files used and created by BCON. There are three sets of sample files delivered with the BCON code in the directory ~/BCON/examples.

agps.wfd This file is the input data for the BCON code. It must be generated as an ASCII file from CAD tools or other geometry codes. It contains unique edge and face definitions for blocks. The detailed format for this file is available in Appendix A.1. That file is also listed in Appendix C.

eagle.jcl, **block.bin**, **bco.dat**, and **relo.dat** are generated by BCON; the first two files are prepared for the EAGLE code and the other two files are for the GMBE code.

eagle.jcl This file provides the EAGLE run stream. It contains: CRAY job control language that specifies the requested CPU size and time, SSD size, compilation information, and file locations; followed by EAGLE statements that define the block size and edge/face definition for every block and block-to-block relationship (i.e. CUT in EAGLE's terminology).

block.bin This file contains EAGLE input data deck. It specifies the block definition in the form of edges and faces. The file is in the CRAY-YMP binary format.

relo.dat This file contains the block-to-block relationships for the six faces of each block and the appropriate data communication between neighboring blocks for the GMBE code calculation. The file format is given in Appendix A.2. The file is also described in Section 4.2.4.

bco.dat This file contains the boundary conditions (i.e. interface, solid surface, inlet, two types of exhaust, and far field) on the six faces or their sub-faces of each block. The file format is given in Appendix A.3. The file is also described in Section 4.2.5.

EAGLE is used to generate the volume grid file named **grid.bin**. It is in PLOT3D binary format on IRIS workstations.

APPENDIX A DETAILED FORMAT DEFINITIONS

A.1 Format definition for the BCON input file, agps.wfd

Card	Column	Format	Symbol	Definition
1	1-10	F10.0	ICOL	Number of points for the edge/sub-edge or number of columns for the face/sub-face.
	11-20	F10.0	IROW	Equals 1 for the edge/sub-edge or number of rows for the face/sub-face.
2	1-10	F10.4	X	x coordinate.
	11-20	F10.4	Y	y coordinate.
	21-30	F10.4	Z	z coordinate.
	31-40	F10.4	X	x coordinate.
	41-50	F10.4	Y	y coordinate.
	51-60	F10.4	Z	z coordinate.

Repeat card two (ICOL + 1) / 2 times. This input preparation will repeat IROW times to accommodate all the points (IROW x ICOL) in the array.

Repeat the cards 1 and 2 preparation for other edges or faces.

Please refer to Appendix C for a sample BCON input file.

A.2 Format definition for the block-to-block relationship input file, relo.dat

Card	Column	Code	Format	Definition
1	1-80		1X	Header card.
2	1-10	NBLK	Free	Block number.
3	1-80		1X	Header card.

4	1-10	NFAC	Free	Block-face number. = 1
	11-20	NREC	Free	Number of records for the block-face 1; this number is equal to number of Card 6.
5	1-80		1X	Header card.
6	1-10	NRA	Free	Record number for the block-face 1.
	11-20	NRB	Free	Record number for the block-face 1 in the neighbor block.
	21-30	NWTP	Free	Record write type. = 1 both local coordinate indices (i.e. J and K) and directions for the block and its neighbor block are same. = 2 local coordinate indices (i.e. J. and K) are same, but directions are opposite for the block and its neighbor block. = 3 local coordinate indices (i.e. J and K) are interchanged, but directions are same for the block and its neighbor block. = 4 both local coordinate indices (i.e. J and K) are interchanged, and directions are opposite for the block and its neighbor block.
	31-40	JMINR	Free	The minimum index J for the record.
	41-50	JMAXR	Free	The maximum index J for the record.
	51-60	KMINR	Free	The minimum index K for the record.
	61-70	KMAXR	Free	The maximum index K for the record.
7	1-80		1X	Header card.
8	1-10	NFAC	Free	Block-face number. = 2

Card	Columns	Variable	Format	Description
	11-20	NREC	Free	Number of records for the block-face 2; this number is equal to number of Card 10.
9	1-80		1X	Header card.
10	1-10	NRA	Free	Record number for the block-face 2.
	11-20	NRB	Free	Record number for the block-face 2 in the neighbor block.
	21-30	NWTP	Free	Record write type. Please see Card 6 for descriptions.
	31-40	JMINR	Free	The minimum index J for the record.
	41-50	JMAXR	Free	The maximum index J for the record.
	51-60	KMINR	Free	The minimum index K for the record.
	61-70	KMAXR	Free	The maximum index K for the record.
11	1-80		1X	Header card.
12	1-10	NFAC	Free	Block-face number. = 3
	11-20	NREC	Free	Number of records for the block-face 3; this number is equal to number of Card 14.
13	1-80		1X	Header card.
14	1-10	NRA	Free	Record number for the block-face 3.
	11-20	NRB	Free	Record number for the block-face 3 in the neighbor block.
	21-30	NWTP	Free	Record write type. = 1 both local coordinate indices (i.e. I and K) and directions for the block and its neighbor block are same. = 2 local coordinate indices (i.e. I. and K) are same, but directions are opposite for the block and its neighbor block.

	31-40	IMINR	Free	The minimum index I for the record.
	41-50	IMAXR	Free	The maximum index I for the record.
	51-60	KMINR	Free	The minimum index K for the record.
	61-70	KMAXR	Free	The maximum index K for the record.
15	1-80		1X	Header card.
16	1-10	NFAC	Free	Block-face number. = 4
	11-20	NREC	Free	Number of records for the block-face 4; this number is equal to number of Card 18.
17	1-80		1X	Header card.
18	1-10	NRA	Free	Record number for the block-face 4.
	11-20	NRB	Free	Record number for the block-face 4 in the neighbor block.
	21-30	NWTP	Free	Record write type. Please see Card 14 for descriptions.
	31-40	IMINR	Free	The minimum index I for the record.
	41-50	IMAXR	Free	The maximum index I for the record.
	51-60	KMINR	Free	The minimum index K for the record.
	61-70	KMAXR	Free	The maximum index K for the record.
19	1-80		1X	Header card.
20	1-10	NFAC	Free	Block-face number. = 5
	11-20	NREC	Free	Number of records for the block-face 5; this number is equal to number of Card 22.

21	1-80		1X	Header card.
22	1-10	NRA	Free	Record number for the block-face 5.
	11-20	NRB	Free	Record number for the block-face 5 in the neighbor block.
	21-30	NWTP	Free	Record write type. = 1 both local coordinate indices (i.e. I and J) and directions for the block and its neighbor block are same. = 2 local coordinate indices (i.e. I. and J) are same, but directions are opposite for the block and its neighbor block.
	31-40	IMINR	Free	The minimum index I for the record.
	41-50	IMAXR	Free	The maximum index I for the record.
	51-60	JMINR	Free	The minimum index J for the record.
	61-70	JMAXR	Free	The maximum index J for the record.
23	1-80		1X	Header card.
24	1-10	NFAC	Free	Block-face number. = 6
	11-20	NREC	Free	Number of records for the block-face 6; this number is equal to number of Card 26.
25	1-80		1X	Header card.
26	1-10	NRA	Free	Record number for the block-face 6.
	11-20	NRB	Free	Record number for the block-face 6 in the neighbor block.
	21-30	NWTP	Free	Record write type. Please see Card 22 for descriptions.
	31-40	IMINR	Free	The minimum index I for the record.

	41-50	IMAXR	Free	The maximum index I for the record.
	51-60	JMINR	Free	The minimum index J for the record.
	61-70	JMAXR	Free	The maximum index J for the record.

Repeat cards 1 through 26 for the next block.

A.3 Format definition for the boundary conditions input file, bco.dat

Card	Column	Code	Format	Definition
1	1-80		1X	Header card.
2	1-10	NBA	Free	Block number.
3	1-80		1X	Header card.
4	1-10	NFA	Free	Block-face number. = 1
	11-20	NBCA	Free	Number of boundary condition patches for the block-face 1; this number is equal to number of Card 6.
5	1-80		1X	Header card.
6	1-10	NTYPE	Free	Boundary condition types for a patch. = 1 interface between blocks = 2 solid surface and symmetry plane = 3 inlet = 4 exhaust one = 5 exhaust two = 6 far field = 7 unused = 8 center line (future option) = 9 downstream facing side on the propeller disk (future option) =10 upstream facing side on the propeller disk (future option)

11-20	JMINB	Free	The minimum index J for the patch.
21-30	JMAXB	Free	The maximum index J for the patch.
31-40	KMINB	Free	The minimum index K for the patch.
41-50	KMAXB	Free	The maximum index K for the patch.

Repeat cards 3 through 6 for the faces 2 through 6. For face 2, assign NFA = 2 on card 4 and use JMINB, JMAXB, KMINB, KMAXB in columns 11-50 of card 6. For face 3, assign NFA = 3 on card 4 and use IMINB, IMAXB, KMINB, KMAXB in columns 11-50 of card 6. For face 4, assign NFA = 4 on card 4 and use IMINB, IMAXB, KMINB, KMAXB in columns 11-50 of card 6. For face 5, assign NFA = 5 on card 4 and use IMINB, IMAXB, JMINB, JMAXB in columns 11-50 of card 6. For face 6, assign NFA = 6 on card 4 and use IMINB, IMAXB, JMINB, JMAXB in columns 11-50 of card 6.

Repeat this input preparation sequence for all blocks.

APPENDIX B PROCEDURE FOR BUILDING THE BCON EXECUTABLE FILE

Thirty-two files are required to install BCON on an IRIS 4D workstation. To bring BCON up on IRIS, a script file 'Makefile' has been prepared to provide the user a quick and simple method to compile and link the program. Modifications to this script file can be made to set up different directory pathnames or input/output.

Since BCON relies on NASA-Ames Research Center's PANEL library for the graphical user interface, the user must first build this library before making BCON executable. Follow the steps described in the 'README' file delivered with the PANEL source to install the library. Then type the UNIX command 'make' in the directory ~/BCON/panel_lib/src. Next run BCON's Makefile by typing the UNIX command 'make' in the directory in which the BCON source is stored, i.e ~/BCON/src_lib. The program will then be compiled and linked to create the executable 'bcon'.

APPENDIX C SAMPLE INPUT FILE FOR BCON

The following data is a sample agps.wfd input file for BCON and is constructed for the example given in Section 5.1. Please refer to figure 5.1. The data is listed in the following sequence: f1 (5 points per row, 11 rows), f2 (5 x 11), f3 (5 x 11), e1 (5 points in a edge), e2 (5 x 1), e3 (5 x 1), e4 (5 x 1), e5 (5 x 1), e6 (5 x 1), e7 (5 x 1), e8 (5 x 1), e9 (11 x 1), e10 (11 x 1).

5.	11.				
.0000	10.0000	10.0000	.0000	7.5000	10.0000
.0000	5.0000	10.0000	.0000	2.5000	10.0000
.0000	.0000	10.0000			
1.0000	10.0000	10.0000	1.0000	7.5000	10.0000
1.0000	5.0000	10.0000	1.0000	2.5000	10.0000
1.0000	.0000	10.0000			
2.0000	10.0000	10.0000	2.0000	7.5000	10.0000
2.0000	5.0000	10.0000	2.0000	2.5000	10.0000
2.0000	.0000	10.0000			
3.0000	10.0000	10.0000	3.0000	7.5000	10.0000
3.0000	5.0000	10.0000	3.0000	2.5000	10.0000
3.0000	.0000	10.0000			
4.0000	10.0000	10.0000	4.0000	7.5000	10.0000
4.0000	5.0000	10.0000	4.0000	2.5000	10.0000
4.0000	.0000	10.0000			
5.0000	10.0000	10.0000	5.0000	7.5000	10.0000
5.0000	5.0000	10.0000	5.0000	2.5000	10.0000
5.0000	.0000	10.0000			
6.0000	10.0000	10.0000	6.0000	7.5000	10.0000
6.0000	5.0000	10.0000	6.0000	2.5000	10.0000
6.0000	.0000	10.0000			
7.0000	10.0000	10.0000	7.0000	7.5000	10.0000
7.0000	5.0000	10.0000	7.0000	2.5000	10.0000
7.0000	.0000	10.0000			
8.0000	10.0000	10.0000	8.0000	7.5000	10.0000
8.0000	5.0000	10.0000	8.0000	2.5000	10.0000
8.0000	.0000	10.0000			
9.0000	10.0000	10.0000	9.0000	7.5000	10.0000
9.0000	5.0000	10.0000	9.0000	2.5000	10.0000
9.0000	.0000	10.0000			
10.0000	10.0000	10.0000	10.0000	7.5000	10.0000
10.0000	5.0000	10.0000	10.0000	2.5000	10.0000
10.0000	.0000	10.0000			

5.	11.				
.0000	10.0000	.0000	.0000	10.0000	2.5000
.0000	10.0000	5.0000	.0000	10.0000	7.5000
.0000	10.0000	10.0000			
1.0000	10.0000	.0000	1.0000	10.0000	2.5000
1.0000	10.0000	5.0000	1.0000	10.0000	7.5000
1.0000	10.0000	10.0000			
2.0000	10.0000	.0000	2.0000	10.0000	2.5000
2.0000	10.0000	5.0000	2.0000	10.0000	7.5000
2.0000	10.0000	10.0000			
3.0000	10.0000	.0000	3.0000	10.0000	2.5000
3.0000	10.0000	5.0000	3.0000	10.0000	7.5000
3.0000	10.0000	10.0000			
4.0000	10.0000	.0000	4.0000	10.0000	2.5000
4.0000	10.0000	5.0000	4.0000	10.0000	7.5000
4.0000	10.0000	10.0000			
5.0000	10.0000	.0000	5.0000	10.0000	2.5000
5.0000	10.0000	5.0000	5.0000	10.0000	7.5000
5.0000	10.0000	10.0000			
6.0000	10.0000	.0000	6.0000	10.0000	2.5000
6.0000	10.0000	5.0000	6.0000	10.0000	7.5000
6.0000	10.0000	10.0000			
7.0000	10.0000	.0000	7.0000	10.0000	2.5000
7.0000	10.0000	5.0000	7.0000	10.0000	7.5000
7.0000	10.0000	10.0000			
8.0000	10.0000	.0000	8.0000	10.0000	2.5000
8.0000	10.0000	5.0000	8.0000	10.0000	7.5000
8.0000	10.0000	10.0000			
9.0000	10.0000	.0000	9.0000	10.0000	2.5000
9.0000	10.0000	5.0000	9.0000	10.0000	7.5000
9.0000	10.0000	10.0000			
10.0000	10.0000	.0000	10.0000	10.0000	2.5000
10.0000	10.0000	5.0000	10.0000	10.0000	7.5000
10.0000	10.0000	10.0000			
5.	11.				
.0000	10.0000	10.0000	.0000	10.0000	12.5000
.0000	10.0000	15.0000	.0000	10.0000	17.5000
.0000	10.0000	20.0000			
1.0000	10.0000	10.0000	1.0000	10.0000	12.5000
1.0000	10.0000	15.0000	1.0000	10.0000	17.5000
1.0000	10.0000	20.0000			
2.0000	10.0000	10.0000	2.0000	10.0000	12.5000
2.0000	10.0000	15.0000	2.0000	10.0000	17.5000

2.0000	10.0000	20.0000			
3.0000	10.0000	10.0000	3.0000	10.0000	12.5000
3.0000	10.0000	15.0000	3.0000	10.0000	17.5000
3.0000	10.0000	20.0000			
4.0000	10.0000	10.0000	4.0000	10.0000	12.5000
4.0000	10.0000	15.0000	4.0000	10.0000	17.5000
4.0000	10.0000	20.0000			
5.0000	10.0000	10.0000	5.0000	10.0000	12.5000
5.0000	10.0000	15.0000	5.0000	10.0000	17.5000
5.0000	10.0000	20.0000			
6.0000	10.0000	10.0000	6.0000	10.0000	12.5000
6.0000	10.0000	15.0000	6.0000	10.0000	17.5000
6.0000	10.0000	20.0000			
7.0000	10.0000	10.0000	7.0000	10.0000	12.5000
7.0000	10.0000	15.0000	7.0000	10.0000	17.5000
7.0000	10.0000	20.0000			
8.0000	10.0000	10.0000	8.0000	10.0000	12.5000
8.0000	10.0000	15.0000	8.0000	10.0000	17.5000
8.0000	10.0000	20.0000			
9.0000	10.0000	10.0000	9.0000	10.0000	12.5000
9.0000	10.0000	15.0000	9.0000	10.0000	17.5000
9.0000	10.0000	20.0000			
10.0000	10.0000	10.0000	10.0000	10.0000	12.5000
10.0000	10.0000	15.0000	10.0000	10.0000	17.5000
10.0000	10.0000	20.0000			
5.	1.				
.0000	.0000	10.0000	.0000	.0000	12.5000
.0000	.0000	15.0000	.0000	.0000	17.5000
.0000	.0000	20.0000			
5.	1.				
10.0000	.0000	20.0000	10.0000	.0000	17.5000
10.0000	.0000	15.0000	10.0000	.0000	12.5000
10.0000	.0000	10.0000			
5.	1.				
.0000	.0000	.0000	.0000	2.5000	.0000
.0000	5.0000	.0000	.0000	7.5000	.0000
.0000	10.0000	.0000			
5.	1.				
10.0000	.0000	.0000	10.0000	2.5000	.0000
10.0000	5.0000	.0000	10.0000	7.5000	.0000
10.0000	10.0000	.0000			
5.	1.				
10.0000	.0000	20.0000	10.0000	2.5000	20.0000

10.0000	5.0000	20.0000	10.0000	7.5000	20.0000
10.0000	10.0000	20.0000			
5.	1.				
.0000	.0000	20.0000	.0000	2.5000	20.0000
.0000	5.0000	20.0000	.0000	7.5000	20.0000
.0000	10.0000	20.0000			
5.	1.				
.0000	.0000	.0000	.0000	.0000	2.5000
.0000	.0000	5.0000	.0000	.0000	7.5000
.0000	.0000	10.0000			
5.	1.				
10.0000	.0000	10.0000	10.0000	.0000	7.5000
10.0000	.0000	5.0000	10.0000	.0000	2.5000
10.0000	.0000	.0000			
11.	1.				
.0000	.0000	20.0000	1.0000	.0000	20.0000
2.0000	.0000	20.0000	3.0000	.0000	20.0000
4.0000	.0000	20.0000	5.0000	.0000	20.0000
6.0000	.0000	20.0000	7.0000	.0000	20.0000
8.0000	.0000	20.0000	9.0000	.0000	20.0000
10.0000	.0000	20.0000			
11.	1.				
.0000	.0000	.0000	1.0000	.0000	.0000
2.0000	.0000	.0000	3.0000	.0000	.0000
4.0000	.0000	.0000	5.0000	.0000	.0000
6.0000	.0000	.0000	7.0000	.0000	.0000
8.0000	.0000	.0000	9.0000	.0000	.0000
10.0000	.0000	.0000			

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.
NASA CR-187484, Volume II

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
A GENERAL MULTIBLOCK EULER CODE FOR PROPULSION INTEGRATION, VOLUME II: USER GUIDE FOR BCON, PRE-PROCESSOR FOR GRID GENERATION AND GMBE

5. Report Date
May 1991

6. Performing Organization Code

7. Author(s)
T. Y. Su, R. A. Appleby, and H. C. Chen

8. Performing Organization Report No.

10. Work Unit No.

9. Performing Organization Name and Address
Boeing Commercial Airplane Group
P.O. Box 3707
Seattle, Wa. 98124-2207

11. Contract or Grant No.
NAS1-18703

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23665-5225

13. Type of Report and Period Covered
Contractor Report

14. Sponsoring Agency Code
535-03-10-01

15. Supplementary Notes
Langley Technical Monitor: Bobby Lee Berrier
Final Report

16. Abstract

BCON is a menu-driven graphics interface program. The BCON input consists of strings or arrays of points generated from a CAD tool or any other surface geometry source. The user needs to design the block topology, prepare the surface geometry definition and surface grids separately. BCON generates input files, that contain the block definitions and the block relationships required for generating a multi-block volume grid with the EAGLE grid generation package. BCON also generates the block boundary conditions file which is used along with the block relationship file as input for the General MultiBlock Euler code (GMBE, Volumes I and III).

17. Key Words (Suggested by Author(s))
Multiblock, Euler Code, Propulsion Integration

18. Distribution Statement
Unclassified - Unlimited
Subject Catagory 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified		

NASA FORM 1626 OCT 86